UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

ESEIAAT
GRETA

AERODYNAMIC STUDY OF A SHIP SAIL INCLUDING CALCULATION OF ITS SHAPE

AUTHOR: MAX AMER VIÑAS
DIRECTOR: LUIS MANUEL PÉREZ LLERA
JUNE 30, 2020

Contents

List of Figures

1 Introduction

1.1 Aim of the project

The aim of the project consists in the aerodynamic study and the calculation of the shape of a ship sail by the means of computational fluid dynamics (CFD) software.

1.2 Scope of the project

The scope and the depth of every area of study are presented below:

- Analyze the history and state of the art of ship sails, their evolution throughout the ages and scientific research behind their behaviour.
- Study of the mathematical models for the calculation of the shape of non-rigid objects under a uniform force and adaptation to the sail shape calculation.
- Create a 3D model of the calculated sail shape using CAD software.
- Creation of a CFD model and mesh to compute a simulation with the adequate parameters, and analyze the results to check for their validity.
- Conclude the project and discuss the results and possible uses of differently shaped sails.

1.3 Project Requirements

1. The designed sail will have a surface of $25m^2$.
2. The aerodynamic study will only consider wind perpendicular to the sail.
3. The simulated wind speed will be of $8m/s$ or $15kn$, a typical speed for intermediate level sailing.

1.4 Background

Throughout history, sailing, and therefore the use of sails, has been instrumental in the development of civilization, allowing humanity to achieve better and faster mobility that on land travel. This has been a tool for commerce, transport and even warfare, as well as for the exploit of marine resources. The earliest documentation of a sailing ship dates back to between 5500 to 5000 B.C. Ancient civilisations and empires like the Egyptians, Romans, Vikings, Greeks, Persians, Chinese and Japanese all used sails to power their ships to keep their empires afloat.

Advances in sailing technology peaked on the 18th century, allowing explorers to take on long voyages to regions with harsh weathers and climates. The designs of the sails evolved from single sailed ships to multi-sailed vessels with triangular and quadrilateral shaped sails for different purposes.

Nowadays, sailing is a recreational activity, as internal combustion engines took over ancient technologies years ago, but the principles of the sail remain the same.

Knowing all the research behind sail technology, CFD is a tool that can help a lot as a way to better understand the behaviour of the air around sails. We have to keep in mind that turbulence has to be taken into account, so many simplifications in the flow might be made as a mean of reducing computation time and making this project viable.

This project will enable us to understand the effect of wind over a sail and also learn the procedure to simulate turbulent flows around an object with CFD software, which will be the critical step to succeed, as CFD simulations are difficult to develop and can easily give an incorrect result that seems plausible, so the discussion of the results is very important.

2 State of the art

2.1 Shape of a non-rigid structure

Non-rigid structures, by definition, are structures that don't have a set form or geometry; they acquire certain shapes reacting to the forces that they receive. The mathematical study of the shapes of non rigid structures is essentially an analysis of the forces applied on a small part of the structure. These small parts can be implemented as differentials, therefore if we analyze the equilibrium of forces applied on these differentials, we can obtain a differential equation, the solution of which is the equation that describes the shape of said structure.

The main force that a structure has to withstand is the force of its own weight. The weight is often uniformly distributed and can be calculated knowing the density of the structure's material and the gravity that applies to it.

The simplest non-rigid structure, and also the first studied, is a rope or chain hanging between two fixed points, deformed by its own weight.

Figure 2.1.1: Chain deformed by its own weight.

In the early 17^{th} century, when Galileo described the trajectory of projectiles without air resistance using the parabola, he also theorized that the shape of a hanging cord would approximate to that of a parabola, noting that this approximation was more accurate as the curvature got smaller.

In 1669, a posthumous publication from Joachim Jungius proved that the shape of a hanging cord didn't follow a parabolic equation, however, he did not find the defining equation.

Later, in 1691, Gottfried Leibniz, Christiaan Huygens, and Johann Bernoulli correctly defined and derived the equation, and their results published that same year.[4]

The formulation of the equation can be derived as following:

Suppose that a cord or cable is suspended between two points A and B, which may be at different heights (2.1.2).

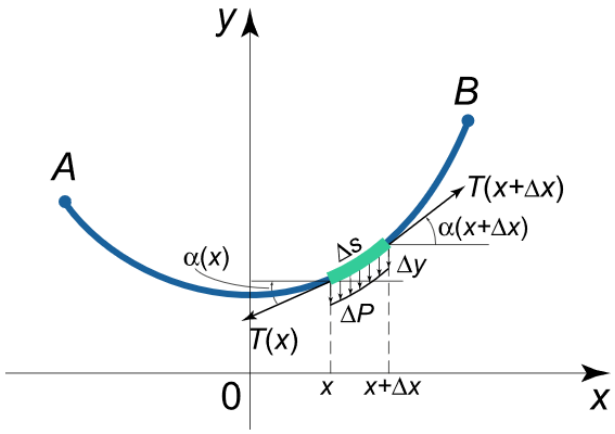


Figure 2.1.2: Diagram of the forces in a hanging cord.[19]

Now consider the equilibrium of forces in a small element of the cord of length ΔS. The forces acting on this element are the uniform force of gravity

$$\Delta P = \rho g A \Delta s \tag{1}$$

where P is the weight of the element, ρ is the density of the cord, A is the surface of the cross section of the element and ΔS is the length of the element, and the tension forces T(x) and T(x+Δx), at the respective points of x and Δx.

The equilibrium of forces in the element projected on the x and y axis, respectively, is written as:

$$T(x+\Delta x)\cdot cos\,\alpha(x+\Delta x) - T(x)\cdot cos\,\alpha(x) = 0 \tag{2}$$

$$T(x+\Delta x)\cdot sin\,\alpha(x+\Delta x) - T(x)\cdot sin\,\alpha(x) - \Delta P = 0 \tag{3}$$

From equation 2 it's known that the horizontal component of the tension is always constant.

$$T(x)\cdot cos\,\alpha(x) = T_0 = const. \tag{4}$$

And equation 3 can be rewritten using differentials.

$$d\left(T(x)\cdot sin\,\alpha(x)\right) = dP(x) \tag{5}$$

And knowing from 4 that $T(x) = \frac{T_0}{cos\,\alpha(x)}$, we have:

$$d\left(T_0\cdot tan\,\alpha(x)\right) = dP(x) \;\Rightarrow\; T_0\cdot d(tan\,\alpha(x)) = dP(x) \tag{6}$$

Now, note that $tan\,(x) = \frac{dy}{dx} = y'$, therefore, the equation of equilibrium can be written in differential form.

$$T_0 \cdot d(y') = dP(x) \Rightarrow T_0 \cdot d(y') = \rho g A ds \tag{7}$$

Following the Pythagorean theorem, the arc section can be simplified as

$$ds = \sqrt{(dx)^2 + (dy)^2} = \sqrt{(dx)^2 \cdot \left(1 + \left(\frac{dy}{dx}\right)^2\right)} = \sqrt{1 + (y')^2} dx \tag{8}$$

which implemented to the differential equation results in:

$$T_0 \cdot \frac{d(y')}{dx} = \rho g A \sqrt{1 + (y')^2} \Rightarrow T_0 \cdot y'' = \rho g A \sqrt{1 + (y')^2} \tag{9}$$

Now the order of this differential equation can be reduced using $z = y'$

$$T_0 \cdot z' = \rho g A \sqrt{1 + (z)^2} \tag{10}$$

and this equation can be solved by separating variables as following:

$$\begin{gathered} T_0 \cdot dz = \rho g A \sqrt{1 + (z)^2} dx \Rightarrow \frac{1}{\sqrt{1+z^2}} dz = \frac{\rho g A}{T_0} dx \Rightarrow \\ \int \frac{1}{\sqrt{1+z^2}} dz = \frac{\rho g A}{T_0} \int dx \Rightarrow ln\left(z + \sqrt{1+z^2}\right) = \frac{x}{\mu} + C_1 \end{gathered} \tag{11}$$

Whera $\mu = \frac{T_0}{\rho g A}$.

Now to calculate C_1 note that the tangent of the curve at its lowest point is horizontal, then:

$$\begin{aligned} y'(x = 0) = z(x = 0) = 0 \\ ln\,1 = 0 + C_1 \Rightarrow C_1 = 0 \end{aligned} \tag{12}$$

Then,

$$z + \left(\sqrt{1+z^2}\right) = e^{\frac{x}{\mu}} \tag{13}$$

To solve this equation, we can multiply both sides by $z - \sqrt{1+z^2}$:

$$\begin{aligned} \left(z + \left(\sqrt{1+z^2}\right)\right)\left(z - \left(\sqrt{1+z^2}\right)\right) = e^{\frac{x}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ z^2 - \left(1+z^2\right) = e^{\frac{x}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ -1 = e^{\frac{x}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ z - \left(\sqrt{1+z^2}\right) = -e^{-\frac{x}{\mu}} \end{aligned} \tag{14}$$

Finally, equations 13 and 14 can be added as following to solve z:

$$\begin{aligned}\left(z+\left(\sqrt{1+z^2}\right)\right)+\left(z-\left(\sqrt{1+z^2}\right)\right) &= e^{\frac{x}{\mu}}+\left(-e^{-\frac{x}{\mu}}\right) \Rightarrow \\ z=\frac{e^{\frac{x}{\mu}}-e^{-\frac{x}{\mu}}}{2} &= sinh\left(\frac{x}{\mu}\right) \Rightarrow \\ y' &= sinh\left(\frac{x}{\mu}\right)\end{aligned} \tag{15}$$

And integrating one more time, the final solution of the differential equation is:

$$y = \mu\, cosh\left(\frac{x}{\mu}\right) \tag{16}$$

2.2 Aerodynamic studies of a sail

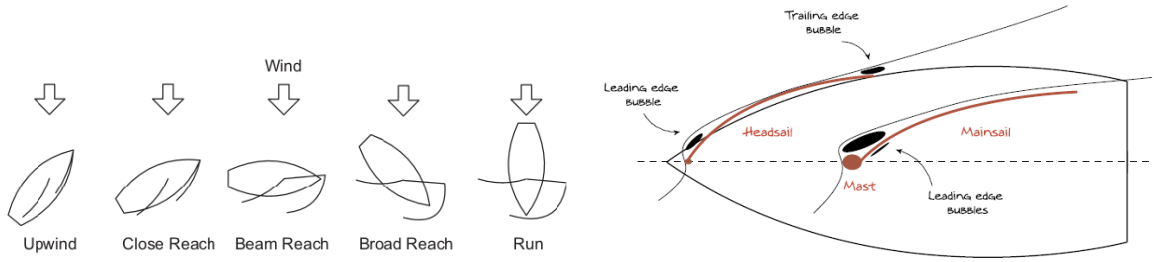
Sails are a clear example of a non-rigid structure interacting with a fluid, therefore a Fluid-Structure interaction: from a structural point of view, they can be defined as thin and flexible tensioned membranes which support aerodynamic loads due to the pressure field around them induced by the wind.

The first historical evidence of humans using sailing boats comes from the Mesopotamian culture, from around 6000 - 4300 BC[21]. Sails have been used throughout all ages as a mean to propel maritime vehicles for very different purposes, and with a great variety of designs and combinations of multiple sails. However, one factor has remained constant up until the modern times: the working principle of a sail has been to use the drag force caused by the wind and translate it to an acceleration to control the motion of the crafts.

However, the research field of the aerodynamics of sails has grown a lot during the last 50 years, by exploiting modern techniques of numerical and experimental investigation, therefore there is an extensive list of literature that can nowadays be found about the topic.

The first studies were conducted with a separate consideration of the structural and fluid dynamic fields, in order to simplify the procedures of those studies. The experimental side was performed inside wind tunnels to recreate the wind environments. As full-size sails have very low thickness, it is impossible to scale geometries while also maintaining the same mechanical properties of the original structure, so sail models are built stiffer than the actual prototypes and sometimes are even directly constructed almost rigid in order to avoid deformation and maintain the shape to correctly study the flow around the sail.

In Izaguirre-Alza[9]'s thesis, the modern sailing techniques are described by the direction of the wind seen by the sail. Nowadays sails are used as wings, in order to take advantage of the wind direction without regard of the direction it is coming from.



(a) Wind directions.[9]

(b) Sail wing.[1]

Figure 2.2.1: Performance of sails in different wind directions and wing effect on a sail.

In this thesis, an in-depth study is made about aerodynamic aspects of sails such as the interaction of the mast with the airflow and the pressure distribution on the sail. As said, experimental data is achieved using a stiffer scale model in order to separate aerodynamic and structural analysis.

Figure 2.2.2: Experimental analysis with a scale model.[9]

Having experimental results allows to compare them to those acquired when executing a numerical analysis through a simulation using a RANS (Reynolds-averaged Navier-Stokes) solver. Further analysis both numerical and experimental note that including the hull to the geometry improves significantly the accuracy of the results.

In recent years, the number of full scale experimental tests has grown, which eliminate the problem of the scaling of the thickness, and the models can be made with real materials to achieve higher accuracy in the results. Results using a full-scale model can be found in Legursky's[10] work in 2012. However, scale models are a lot more useful in many cases primarily because their lower cost and also because combining experimental data from scale models and numerical analysis, the results obtained are already accurate. A good example of the combination of scale models and numerical can be found in Bayati, Muggiasca and Vandone's[1] work published in 2019, in which numerical models are validated comparing them with experimental results in a scale model, in order to obtain useful data from the simulations that is difficult to acquire from experimental analysis.

In the present, the most innovative research is directed towards the use of rigid wings, with airplane airfoils, placed vertically on the deck of large vessels as a mean to assist to their propulsion and increase their efficiency, thus reducing operational costs. As studied in a Polish Maritime Research paper[11], a numerical analysis is conducted on this vertical wing system, and the results are compared to numerical results of more traditional sails obtained through numerical methods verified with experimental data, in order to compare the performance and viability of the new system.

Regarding to the applicability of numerical simulations to the design of a sail,the first approaches used non-viscous fluid-dynamic software to run the simulations. As computational power has increased in recent years, studies moved on to viscous RANS solvers, which allowed to evaluate more complex flows. One of the first numerical approaches to the design of sailing yacht righs[3] was done in 1991 where a software is implemented to calculate the aerodynamic forces. Another interesting study by Ghelardi, Freda, Rizzo and Villa[7] analyzes the behaviour of a square sail hanging from a beam undergoing large displacements due to the aerodynamic forces. That study specifies how it is important to choose a simple geometry when developing a new method or solver.

2.3 CFD simulation examples

CFD simulations are very useful when studying and trying to understand the external flow of fluids around objects. Some useful information when designing a simulation can be found in the openFoam user guide[6], which contains all the basic information and guides the user through all the software's utilities. This includes different solvers, different boundary conditions, different utilities as well as providing some tutorials and information in order to help the user manage his way through the development of a case. There are many studies that work with external flow, but the main example is the tutorial *wind around buildings* provided by the software openFoam, which has concrete procedures and conditions that allow to simulate the wind flow around a group of buildings. Segersson's work[17] (2017) provides an in-depth explanation of the important steps to follow when using openFoam, how to set up boundary conditions and how to apply and evaluate wall-functions for the atmospheric boundary layer. It includes also an explanation of the theory behind the simulation, how to model ground roughness with boundary conditions and how to represent porous volumes with porosity models. Finally, it explains how to implement the simulation, how to control it, adjust run-time and how to adjust different roughness lengths within a wall.

Then, in another paper by Sundararaj S., Mohan R. and Thiagarajan K.[13] the computational methodology is explained, and the results are discussed and explained, including streamlines, velocity and pressure distributions, and the analysis of vortices and postprocessing features such as drag calculations.

3 Analysis and definition

3.1 Adaptation and concretion of shape calculation methods

In order to develop an initial approach to this problem, some simplifications can be made in order to be able to perform a load analysis similar to the shape of a non-rigid structure problem 2.1.2. In this problem, the weight of the cord is parallel to the vertical axis, resulting in a constant value of the horizontal forces. As the pressure forces are always perpendicular to the surface they are applied to, the same analysis would result in a pressure load that varies with the angle of the surface, which results in a changing value of the horizontal force that renders useless the described procedure.

Therefore, the first simplification is to use a virtually flat surface, with a curvature small enough that the pressure force can be considered parallel to one axis. This simplification is reasonable, as we can assume the sail to be a flat sheet of fabric between two bars, where the fabric tension is actually zero but the sail stays in a flat shape because the fabric length is exactly the distance between the top and bottom bars.

(a) Pressure force on a curved sail.

(b) Pressure force on a virtually flat sail.

Figure 3.1.1: Pressure force diagrams with different sail shapes.

The second simplification is to suppose body forces to be negligible. This simplification can be justified through some values seen in Ghelardi et al.[7]. In this paper, the tested fabric has a density of $\rho = 235 g/m^2$ and the results for a square sail of 330x330 mm show a total drag of approximately 10N with an incident wind of 3.7 m/s. The total weight of the sail is approximately 0.25N. This 40:1 ratio increases when studying faster flows, due to the increase in pressure over the sail. As shown, the drag forces are more than an order of magnitude higher than the body forces when using a generic sail fabric, therefore body forces can be neglected.

The third simplification is to suppose drag as a uniform load equally distributed along the sail surface.

This simplification does not exactly resemble reality, as the drag force is a consequence of the difference in pressure on the surface of the sail, but as the pressure distribution on the surface of the sail is unknown beforehand, the supposition of a uniform distribution is made to proceed with the first step of the calculation.

Some minor considerations will also be made to simplify the calculations by supposing ideal conditions such as constant air density and temperature at sea level, which is not the real case as density changes with pressure, but the incoming flow conditions will be used as reference. Also, the drag coefficient can only be calculated through the total drag, and as we will not have the total drag value before performing the numerical simulation, the drag coefficient will be taken from experimental data of flat plates, which is acceptable as the surface differentials on the sail will essentially be flat plates.

3.2 Adaptation and concretion of the aerodynamic study

The aerodynamic study of the sail has some very clear objectives listed below:

- Obtain the velocity field around the sail, in order to view possible high velocity zones.
- Obtain the pressure field around the sail, in order to calculate total drag in postprocessing.
- Obtain the streamlines around the sail, in order to visualize vorticity and turbulence.

The study will be purely numerical and it will be performed in conjunction with said numerical simulation and the results obtained from it. Results and data will be analysed and discussed and the discussion as well as their effect on the initial conditions and simplifications.

The methodology will follow by using the calculated shape to model a 3d solid, and analysing it as a rigid structure, knowing that the shape is already a stable form of the sail. The results obtained from this simulation will be analog to the flow around a stable non-rigid structure.

3.3 Adaptation and concretion of CFD

The numerical simulation will be performed with the OpenFOAM software following the steps of the *WindAroundBuildings*. The parameters of the simulation will be considered in order to be more accurate for the external flow around an object. In order to understand and choose between those parameters, specific documentation will be analyzed, such as OpenFOAM's user guide[6] and specific CFD forums where users post information to understand and resolve other user's cases and specific flaws.

4 Case study

4.1 Calculation of the shape

Parting from the simplifications presented in section 3.1, consider the equilibrium of forces in a small element of the sail of surface ΔS. The forces acting on this element are the uniform load of wind drag

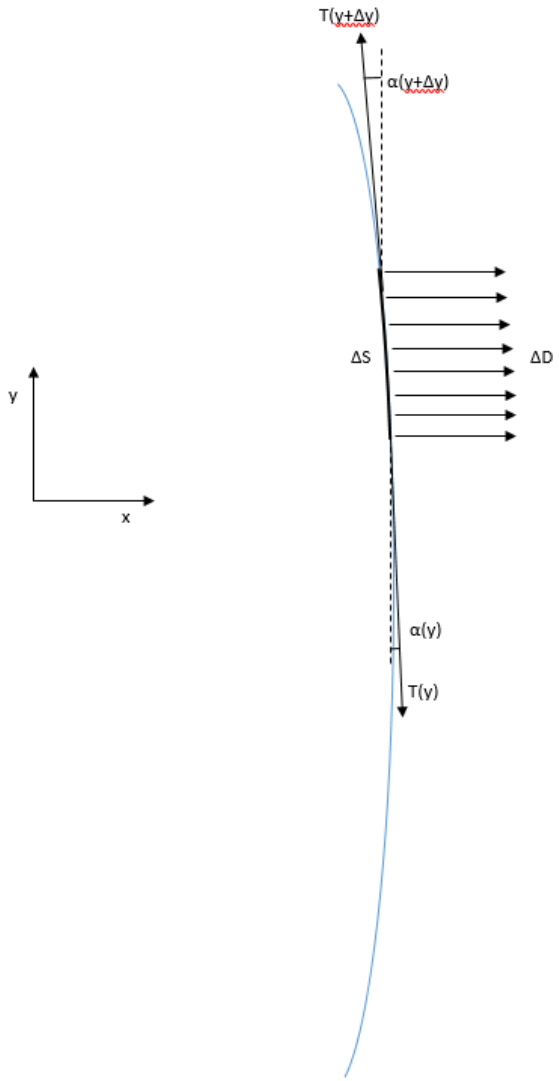


Figure 4.1.1: Diagram of the forces in a *virtually* flat sail.

This wind load can be described as

$$\Delta D = \frac{1}{2}\rho v^2 C_D \Delta S \tag{17}$$

where D is the drag of the element, ρ is the density of the air flowing around the sail, v is the free stream velocity of the air, C_D is the drag coefficient of a flat plate perpendicular to the flow and ΔS is the surface of the element (which can be decomposed as $\Delta S = \Delta s \cdot l$, where Δs is the arc length and l is the sail width). Also, the tension forces T(y) and T(y+Δy), at the respective points of y and $y + \Delta$y are

represented too.

The equilibrium of forces in the element projected on the y and x axis, respectively, is written as:

$$T(y+\Delta y)\cdot cos\,\alpha(y+\Delta y)-T(y)\cdot cos\,\alpha(y)=0 \tag{18}$$

$$-T(y+\Delta y)\cdot sin\,\alpha(y+\Delta y)+T(y)\cdot sin\,\alpha(y)-\Delta D=0 \tag{19}$$

From equation 18 it's known that the vertical component of the tension is always constant.

$$T(y)\cdot cos\,\alpha(y)=T_0=const. \tag{20}$$

And equation 19 can be rewritten using differentials.

$$d\left(T(y)\cdot sin\,\alpha(y)\right)=dD(y) \tag{21}$$

And knowing from 20 that $T(y)=\frac{T_0}{cos\,\alpha(y)}$, we have:

$$d\left(T_0\cdot tan\,\alpha(y)\right)=dD(y)\;\Rightarrow\;T_0\cdot d(tan\,\alpha(y))=dP(y) \tag{22}$$

Now, note that $tan\,(y)=\frac{dx}{dy}=x'$, therefore, the equation of equilibrium can be written in difrerential form.

$$T_0\cdot d(x')=dP(y)\;\Rightarrow\;T_0\cdot d(y')=\frac{1}{2}\rho v^2 C_D l ds \tag{23}$$

Following the Pythagorean theorem, the arc section can be simplified as

$$ds=\sqrt{(dy)^2+(dx)^2}=\sqrt{(dy)^2\cdot\left(1+\left(\frac{dx}{dy}\right)^2\right)}=\sqrt{1+(x')^2}dy \tag{24}$$

which implemented to the differential equation results in:

$$T_0\cdot\frac{d(x')}{dy}=\frac{1}{2}\rho v^2 C_D l\sqrt{1+(x')^2}\;\Rightarrow\;T_0\cdot x''=\frac{1}{2}\rho v^2 C_D l\sqrt{1+(x')^2} \tag{25}$$

Now the order of this differential equation can be reduced using $z=x'$

$$T_0\cdot z'=\frac{1}{2}\rho v^2 C_D l\sqrt{1+(z)^2} \tag{26}$$

and this equation can be solved by separating variables as following:

$$\begin{aligned}
T_0\cdot\frac{dz}{dy}&=\frac{1}{2}\rho v^2 C_D l\sqrt{1+(z)^2}\\
\frac{1}{\sqrt{1+z^2}}dz&=\frac{\frac{1}{2}\rho v^2 C_D l}{T_0}dy\\
\int\frac{1}{\sqrt{1+z^2}}dz&=\frac{\frac{1}{2}\rho v^2 C_D l}{T_0}\int dy
\end{aligned}$$

Now substitute $z = tan(u)$ and $dz = \frac{1}{cos^2(u)}du$

$$\int \frac{1}{\sqrt{1+tan^2(u)}} \frac{du}{cos^2(u)} = \frac{y}{\mu} + C_1$$

$$where\, \mu = \frac{T_0}{\frac{1}{2}\rho v^2 C_D l}$$

$$\int \frac{1}{\sqrt{\frac{1}{cos^2(u)}}} \frac{du}{cos^2(u)} = \frac{y}{\mu} + C_1$$

$$\int \frac{1}{cos(u)} du = \frac{y}{\mu} + C_1$$

Multiply numerator and denominator by $tan(u) + \frac{1}{cos(u)}$

$$\int \frac{tan(u) + \frac{1}{cos(u)}}{cos(u)\left(tan(u) + \frac{1}{cos(u)}\right)} du = \frac{y}{\mu} + C_1$$

$$\int \frac{\frac{1+sin(u)}{cos^2(u)}}{\frac{1+sin(u)}{cos(u)}} du = \frac{y}{\mu} + C_1$$

Substitute $s = \frac{1+sin(u)}{cos(u)}$ and $ds = \frac{1+sin(u)}{cos(u)}du$

$$\int \frac{1}{s} ds = \frac{y}{\mu} + C_1$$

$$ln(s) = \frac{y}{\mu} + C_1$$

And now, undoing the substitutions

$$ln\left(\frac{1+sin(u)}{cos(u)}\right) = \frac{y}{\mu} + C_1$$

$$ln\left(\frac{1+sin(arctan(z))}{cos(arctan(z))}\right) = \frac{y}{\mu} + C_1$$

Which can be simplified as

$$ln\left(z + \sqrt{(z^2+1)}\right) = \frac{y}{\mu} + C_1 \tag{27}$$

Knowing $\mu = \frac{T_0}{\frac{1}{2}\rho v^2 C_D l}$.

In order to obtain the final solution, C_1 can be calculated knowing that the tangent of the curve is vertical at the middle of the sail:

$$\begin{aligned} x'(y=0) = z(y=0) = 0 \\ ln\,1 = 0 + C_1 \Rightarrow C_1 = 0 \end{aligned} \tag{28}$$

Then,

$$z + \left(\sqrt{1+z^2}\right) = e^{\frac{y}{\mu}} \tag{29}$$

Multiplying both sides by $z - \sqrt{1+z^2}$ as follows:

$$\begin{aligned} \left(z + \left(\sqrt{1+z^2}\right)\right)\left(z - \left(\sqrt{1+z^2}\right)\right) = e^{\frac{y}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ z^2 - \left(1+z^2\right) = e^{\frac{y}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ -1 = e^{\frac{y}{\mu}}\left(z - \left(\sqrt{1+z^2}\right)\right) \Rightarrow \\ z - \left(\sqrt{1+z^2}\right) = -e^{-\frac{y}{\mu}} \end{aligned} \tag{30}$$

Finally, equations 29 and 30 can be added as following to solve z:

$$\begin{aligned} \left(z + \left(\sqrt{1+z^2}\right)\right) + \left(z - \left(\sqrt{1+z^2}\right)\right) = e^{\frac{y}{\mu}} + \left(-e^{-\frac{y}{\mu}}\right) \Rightarrow \\ z = \frac{e^{\frac{y}{\mu}} - e^{-\frac{y}{\mu}}}{2} = sinh\left(\frac{y}{\mu}\right) \Rightarrow \\ x' = sinh\left(\frac{y}{\mu}\right) \end{aligned} \tag{31}$$

And integrating one more time, the final solution of the differential equation is:

$$\begin{aligned} x = \mu\, cosh\left(\frac{y}{\mu}\right) \\ \mu = \frac{T_0}{\frac{1}{2}\rho v^2 C_D l} \end{aligned} \tag{32}$$

This result is analog to the deformation of a hanging cord due to the nature of the case and the simplifications in the design.Now, the value of μ can be determined in order to specify the equation.

The air density at sea level is $\rho = 1.225 kg/m^3$, the velocity of the fluid is $v = 8 m/s$, the drag coefficient of a square flat plate is $C_D = 1.17$ [5] and the length l is 5 meters.

In order to calculate the value of T_0, the mechanical properties of the dacron sailcloth (the most common material used in sailing rigs) are key to determine the force that maintains the equilibrium state.

These mechanical properties are obtained from the FSI case study [7], where a series of tensile tests are performed in order to determine them.

As sailcloths are made from woven fibers, the properties used will be those of the weakest direction. It is known from the results of those tests that the weakest direction of the sailcloth is the bias direction

(45º angle respect to the fill and bias directions), which has a maximum stress on the elastic region of $\sigma = 8.7N/mm^2$.

It is known that $\sigma = \frac{F}{A}$, therefore, the cross sectional area of the sail is needed to calculate the tension. Knowing the values of weight per squared meter of the weave, and also the density of the dacron, it is possible to calculate the thickness of the fabric.

$$\rho = 1.38g/cm^3$$
$$\sigma = 235g/m^2$$

Note that sigma here is referring to the area density.

Then,

$$Thickness = e = \frac{\rho}{\sigma} = \frac{235g/m^2}{1380000g/m^3}$$
$$e = 0.17mm$$

Following the stress definition, and knowing the dimensions of the sail, it is possible to calculate the tension to which the sailcloth is subjected:

$$\sigma = \frac{T_0}{A}$$
$$T_0 = \sigma \cdot e \cdot l \tag{33}$$

Implementing equation 33 into the μ calculation the final numeric result is obtained.

$$\mu = \frac{\sigma \cdot e}{\frac{1}{2}\rho v^2 C_D}$$
$$\mu = 32.248$$

$$x = 32.248\, cosh\left(\frac{y}{32.248}\right) \tag{34}$$

Which can be seen in the following graph (figure 4.1.2).

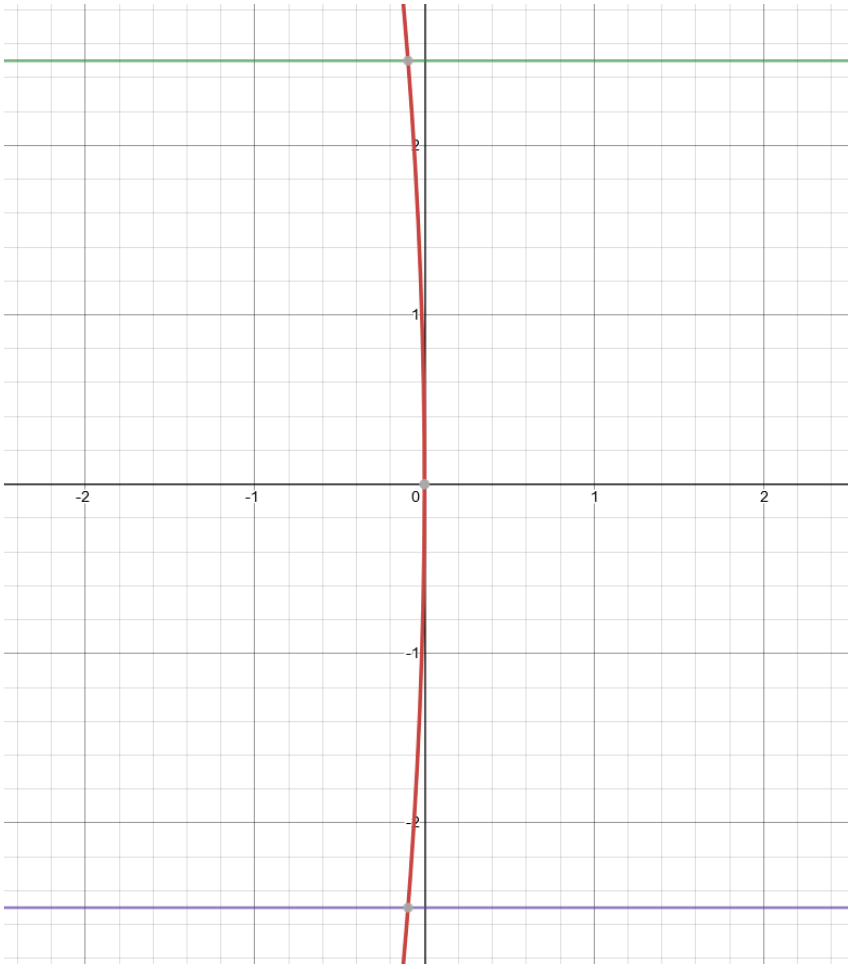


Figure 4.1.2: Graph of the resulting equation.

The plotted equation is $x = -32.248\,cosh\left(\frac{y}{32.248}\right) + 32.248$ which is just a mirrored and relocated sample but maintains the same shape, just to make it clearer on the graph and match it to the case geometry.

As it can be seen, the equation is bounded between y = 2.5 and y = -2.5, as the height of the sail is 5m. The maximum horizontal displacement of the sail is 0.097 m.

A commentary on this equation, as it was described in the state of the art (section 2.1), this type of shapes can be approximated to parabolic equations on their highest curvature interval, as it can be seen in figure 4.1.3. In this case, the resulting equation can be approximated as $x = -\frac{y}{8.02893}^2$.

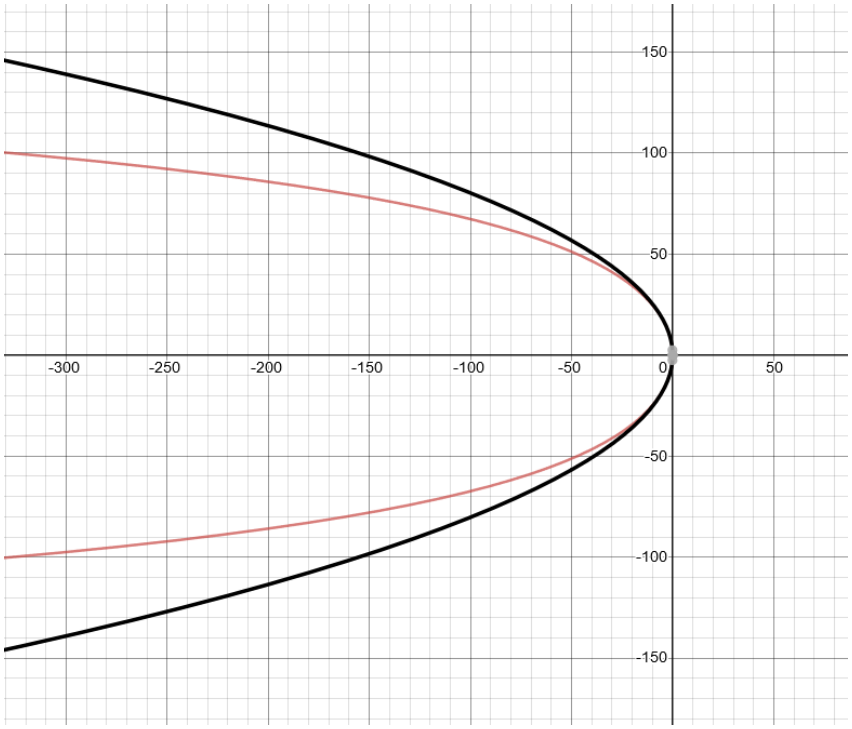


Figure 4.1.3: Parabolic (in black) approximation to a hyperbolic cosine (in red).

As it can be seen, these two functions have a similar shape and are almost identically on their maximum curvature interval.

In order to check the validity of the obtained result, the strain of the sail can be calculated and compared to the strain of dacron for a $\sigma = 8.7N/mm^2$ [7].

The original length of the sail is a known value, therefore the only procedure needed is to calculate the arc length of the deformed sail.

To calculate the length of a continuous function on the interval [a,b], said function can be divided into n sub-intervals of length Δy thus denoting a set of points P_i. The length of the function can be approximated as the sum of all straight lines connecting the points P_i. The length of each straight line is $|P_i - P_{i-1}|$, therefore:

$$L \approx \sum_{i=1}^{n} |P_i - P_{i-1}|$$

Now make n infinitely large so the sum of lengths of the straigt lines will be equal to the function length

$$L = \lim_{n\to\infty} \sum_{i=1}^{n} |P_i - P_{i-1}|$$

The length of each straight line can be rewritten as

$$|P_i - P_{i-1}| = \sqrt{\Delta y^2 + \Delta x^2}$$

And rearranging terms

$$L = \lim_{n\to\infty} \sum_{i=1}^{n} \sqrt{1 + \left(\frac{\Delta x}{\Delta y}\right)^2}\, \Delta y$$

As $\frac{\Delta x}{\Delta y}$ is the derivative of the function x, and the infinite sum is the definition of the integral, the exact length of the curve is:

$$L = \int_a^b \sqrt{1 + (x')^2}\, dy$$

The sail shape function is the input x, and $x' = sinh\left(\frac{y}{32.248}\right)$, as well as an upper and lower limit of ± 2.5. Therefore, the final length of the sail is:

$$L = \int_{-2.5}^{2.5} \sqrt{1 + sinh^2\left(\frac{y}{32.248}\right)}\, dy \tag{35}$$

This integral is easily solved through the hyperbolic identity $cosh^2(y) - sinh^2(y) = 1$, so it's known that $1 + sinh^2(y) = cosh^2(y)$.

$$L = \int_{-2.5}^{2.5} cosh\left(\frac{y}{32.248}\right) dy$$

$$L = 5.00501m$$

Which corresponds to a strain of

$$\varepsilon = \frac{0.00501}{5} = 0.001$$

Comparing this result with the stress-strain results on the FSI study[7] seen on figure 4.1.4, it's visible that the results don't match the bias curve, for a stress of $8.7 N/mm^2$, the strain should be around 0.005.

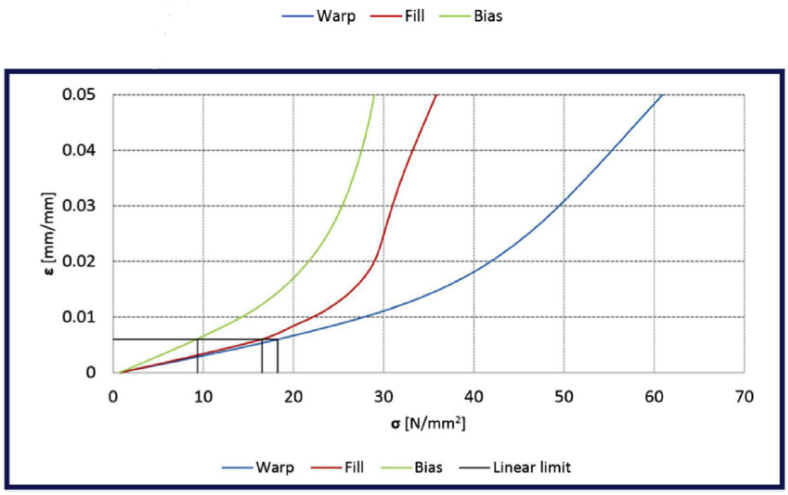


Figure 4.1.4: Stress-strain results of the tensile test of the dacron sailcloth.[7]

As this result does not comply with the equilibrium condition of the sail, a relation must be established between the stress-strain values and the equation of the shape of the sail. As the equation to calculate ε is already described, it can be equaled to the linear equation of strain, obtained from the definition of Young's modulus, a value provided in this cited article.

The Young's modulus is defined as $E = \frac{\sigma}{\varepsilon}$, and for the bias direction, $E = 1583.23N/mm^2$, therefore:

$$\varepsilon = \frac{1}{E}\sigma$$

And from the sail shape calculation:

$$\varepsilon = \frac{L - L_0}{L_0}$$

$$\varepsilon = \frac{\int_{-2.5}^{2.5} cosh\left(\frac{y}{\mu}\right) dy - L_0}{L_0}$$

Now the integral can be written as its result

$$\varepsilon = \frac{2\mu sinh\left(\frac{2.5}{\mu}\right) - L_0}{L_0}$$

By matching the two strain equations we obtain:

$$\frac{1}{E}\sigma = \frac{2\mu sinh\left(\frac{2.5}{\mu}\right) - L_0}{L_0} \tag{36}$$

where $\mu = \frac{\sigma \cdot e}{\frac{1}{2}\rho v^2 C_D}$.

Now the only unknown value is σ, which by resolving the equation we obtain:

$$\sigma = 4.9345N/mm^2$$

With this final value of stress, μ can be recalculated to $\mu = 18.29$, which modifies the sail shape as such.

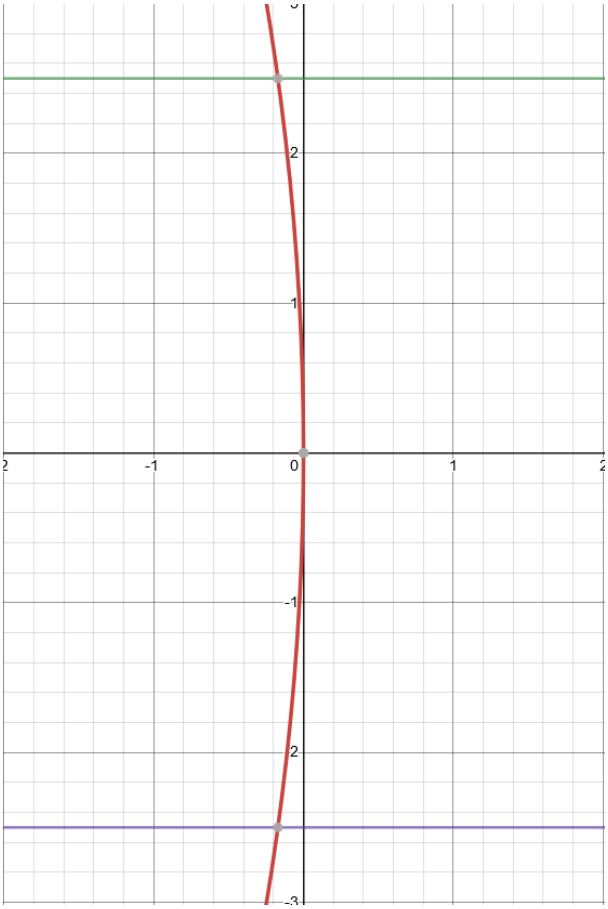


Figure 4.1.5: Graph of the sail shape.

This recalculated shape has a maximum horizontal displacement of 0.171m. The final length of the sail is $L = 5.0156m$ with an associated strain of $\varepsilon = 0.00312$ which corresponds to the value of $\sigma - \varepsilon$ from figure 4.1.4.

4.2 3D sail design

The 3D CAD shape will be designed using the software Salome-Meca, which includes a meshing utility useful after the shape is constructed. As the shape equation is already known, in order to model the 3D structure, a rectangular sail section will be extruded along a line following said shape equation as seen in figure 4.2.1.

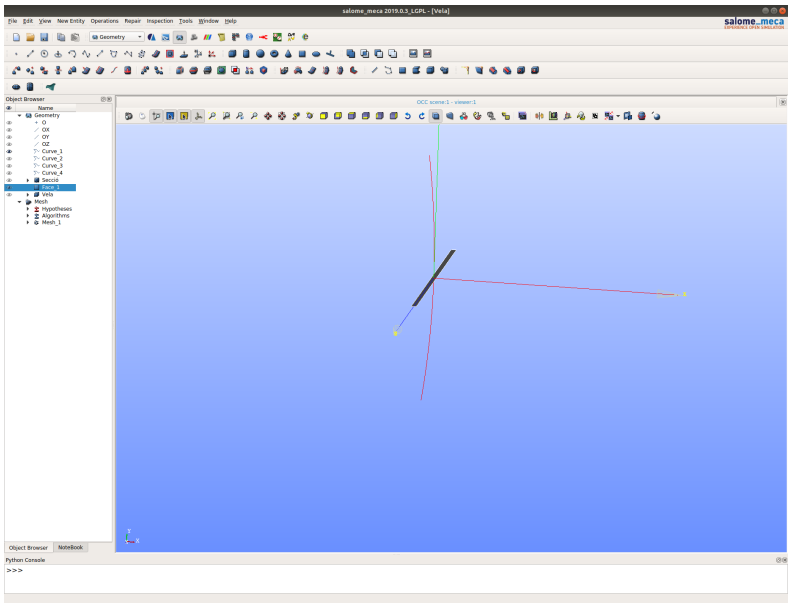


Figure 4.2.1: Rectangular section and extrusion line.

Once the section is extruded, a 3D solid is obtained. For the aerodynamic simulation, the solid sail is not useful, as the intended area of study is the flow around the sail. Therefore, only the surface geometry of the sail is needed to bound the movement of the fluid. In order to obtain that, the solid sail is converted into a shell before starting with the meshing procedure.

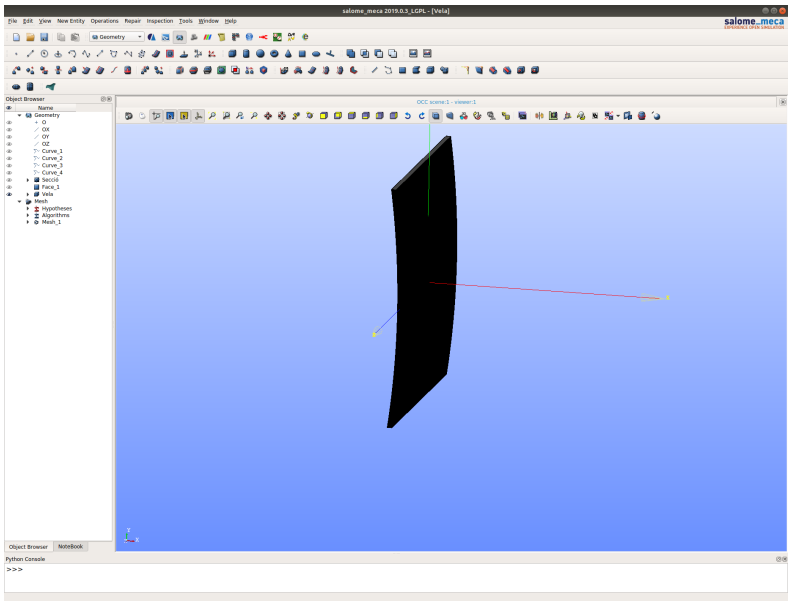


Figure 4.2.2: Final shell of the sail.

After the shell is obtained, it can be transformed to a surface mesh of the sail, using the meshing feature of Salome-Meca. The mesh is created using automatic 2D triangulation of the shell, and a maximum length division of 0.01, then the mesh is computed and the resulting mesh can be exported as an STL file to be used with the simulation software OpenFoam. (Note in figure 4.2.3 that the sail appears completely white because the mesh is very fine).

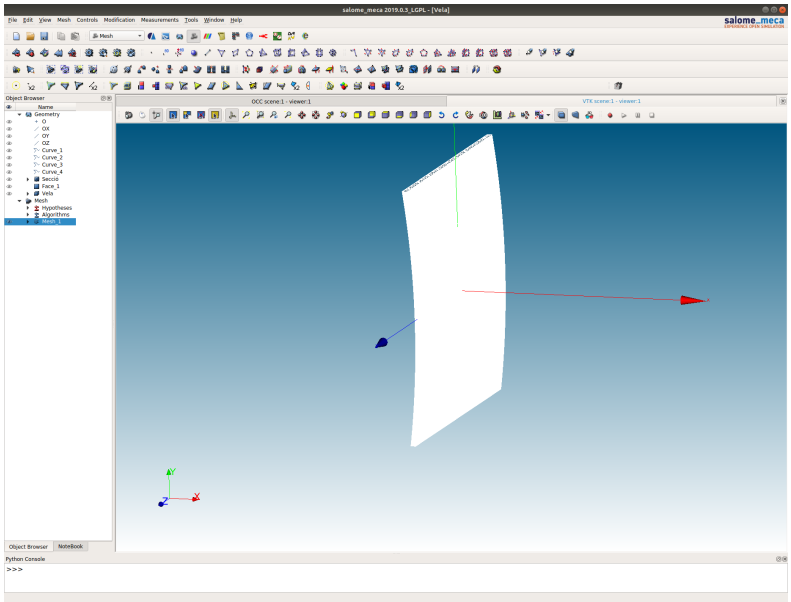


Figure 4.2.3: Final mesh of the sail.

4.3 Meshing and boundary conditions

4.3.1 Meshing with OpenFoam

As it has already been stated, the simulation will focus on the flow of air around the sail, therefore the desired mesh has to envelop the sail geometry, and must be big enough to be able to observe the velocity and pressure fields.

The first step will be to create a bounding box which will include the outer boundaries of the mesh, using the feature *blockMesh*, which can be seen in annex A. The sail mesh previously created will have to fit inside this bounding box, therefore the first condition to take into account will be the sail mesh dimensions. The sail's maximum and minimum x,y and z positions respectively are +0.271/0, +2.5/-2.5 and +2.5/-2.5. Knowing this, the bounding box dimensions can be set. This mesh design will set the bottom of the sail in contact with the bottom of the bounding box, therefore it will simulate a sail attached to the floor. Regarding the length and height, the only restriction is to give enough room before and after the sail and upwards and sideways of its perimeter. The final bounding box dimensions selected are +30/-30, +30/-2.5 and +20/-20.

Also, along with the mesh dimensions, all the boundary regions and functions have to be set. In this case, each region is defined by its bounding vertices, and catalogued according to their type. The resulting regions with their respective types are:

- Inlet, type patch
- Outlet, type patch
- Floor, type wall
- Air, type wall

The resulting mesh can be seen in the following figure.

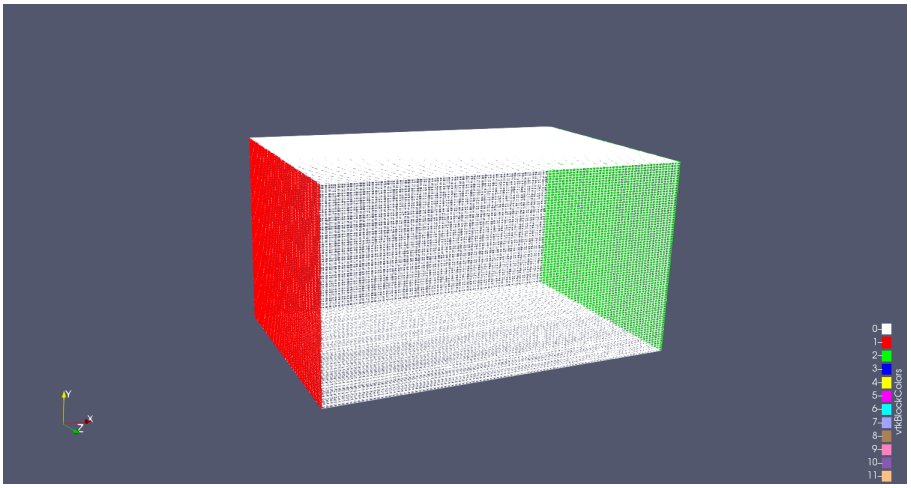
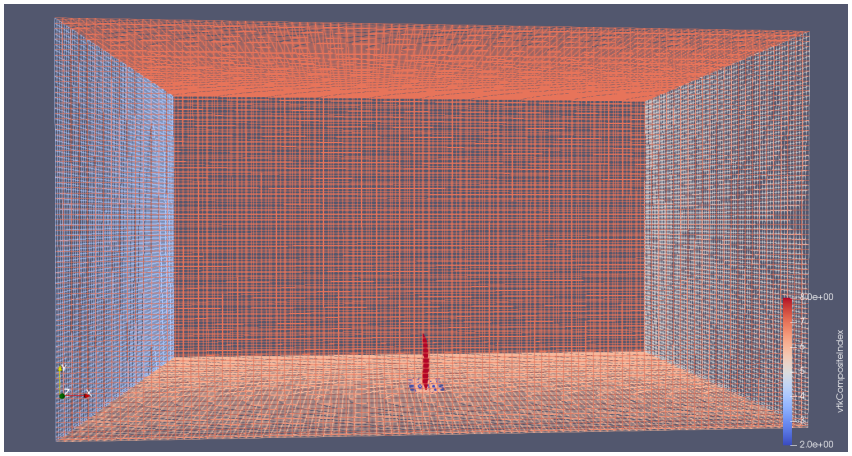


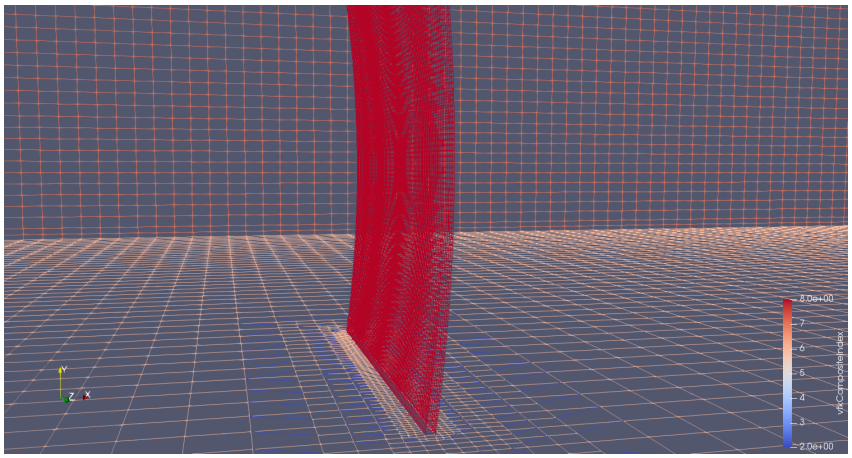
Figure 4.3.1: Resulting blockMesh.

As it's noticeable, this mesh does not include the sail geometry mesh previously generated. In order to merge the bounding box with the sail, the feature *snappyHexMesh*, which can be seen in annex B, is used. This utility reads the sail mesh file and the blockMesh, as well as some parameters such as the

level of refinement desired around the sail and the quality of the resulting mesh, and generates a final mesh as an intersection of both input files. The final mesh can be seen in the following figure.



(a) Full view.



(b) Detail view.

Figure 4.3.2: Final mesh of the fluid around the sail.

4.3.2 Boundary conditions

Once the mesh geometry is totally defined, the boundary conditions must be set in order to proceed with the simulation. The selection of boundary conditions is very important, because if the case is not set correctly, the solver will most likely crash, or the given results will be incoherent. In this case, the mesh consists of an inlet and an outlet, the floor and the sail, which will be treated as a wall, and the air, which will be set to not interact with the flow, in order to eliminate the `tube` simulation.

The two main boundary conditions to be set are the velocity and the pressure.

For the velocity U (see annex C) the following boundary conditions are set:

- **Inlet:** The inlet has a *fixedValue* condition of uniform (8, 0, 0) m/s.
- **Outlet:** The outlet has a *pressureInletOutletVelocity* condition with an initial value of (0, 0, 0), which sets the velocity at the outlet as the result of the simulation, maintaining it always positive or equal to zero (prevents backflow).

- **Floor:** The floor has a *noSlip* condition to simulate the friction of the fluid with a surface.
- **Sail (vela):** The sail also has a *noSlip* condition to simulate the friction of the fluid with a surface.
- **Air:** The air has a *slip* condition to simulate the continuity of the atmosphere and not limit the simulation to a tube.

The rest of the mesh is set to have an initial value (before computing) of uniform 0 m/s.

For the pressure P (see annex D) the following boundary conditions are set:

- **Inlet:** The inlet has a *zeroGradient* condition as the pressure is supposed to be constant at the inlet.
- **Outlet:** The outlet has a *fixedValue* condition equal to 0, as the pressure at the outlet is equal to the atmospheric pressure.
- **Floor:** The floor also has a *zeroGradient* condition.
- **Sail (vela):** The sail also has a *zeroGradient* condition.
- **Air:** The air also has a *zeroGradient* condition.

The rest of the mesh is set to have an initial value (before computing) of uniform 0 N/m^2.

The secondary boundary conditions that must be set are files needed in order to compute the RAS solvers. The solvers studied for the simulations are the *kEpsilon* model, *kOmegaSST* model and the *SpalartAllmaras* model. These solvers include the files of epsilon, k, nut, nutilda and omega (respectively, annexes E, F, G, H and I).

The conditions set for each variable are:

- epsilon
  - **Inlet:** *fixedValue* condition.
  - **Outlet:** *inletOutlet* condition.
  - **Floor:** *epsilonWallFunction* condition.
  - **Sail (vela):** *epsilonWallFunction* condition.
  - **Air:** *epsilonWallFunction* condition.
- k
  - **Inlet:** *fixedValue* condition.
  - **Outlet:** *inletOutlet* condition.
  - **Floor:** *kqRWallFunction* condition.
  - **Sail (vela):** *kqRWallFunction* condition.
  - **Air:** *kqRWallFunction* condition.
- omega
  - **Inlet:** *inletOutlet* condition.
  - **Outlet:** *inletOutlet* condition.

  - **Floor:** *omegaWallFunction* condition.
  - **Sail (vela):** *omegaWallFunction* condition.
  - **Air:** *omegaWallFunction* condition.

- nut
  - **Inlet:** *freeStream* condition.
  - **Outlet:** *freeStream* condition.
  - **Floor:** *nutUSpaldingWallFunction* condition.
  - **Sail (vela):** *nutUSpaldingWallFunction* condition.
  - **Air:** *nutUSpaldingWallFunction* condition.

- nutilda
  - **Inlet:** *freeStream* condition.
  - **Outlet:** *freeStream* condition.
  - **Floor:** *fixedValue* condition.
  - **Sail (vela):** *fixedValue* condition.
  - **Air:** *fixedValue* condition.

These variables are used to calculate turbulent flow in turbulence models. The initial values of each variable for their respective condition (epsilon - *fixedValue*, k - *fixedValue*, nut - *freeStream* and nuTilda - *freeStream*) are estimated using formulas for computing turbulence model variables[14] [15]. For each variable, then:

$$k = \frac{3}{2}(I \cdot U)^2 = 0.04 \tag{37}$$

Where I is the turbulence intensity, in this case the value is set to 2 %.

$$\varepsilon = C_\mu \frac{k^{1.5}}{l} = 0.00028 \tag{38}$$

Where l is the turbulent length scale, which is a physical quantity describing the size of the large energy-containing vortices in a turbulent flow. In this case, this dimention is set to half the sail height, 2.5 m, and C_μ is a turbulent constant with a value of 0.09.

$$\omega = \frac{\sqrt{k}}{l} = 0.08 \tag{39}$$

$$\tilde{\nu} = \sqrt{\frac{3}{2}} U \cdot I \cdot l = 0.49 \tag{40}$$

$$\nu = 0.49 \tag{41}$$

The value of nut is considered to be the same order of magnitude as the nuTilda, and for the purpose of obtaining a converging simulation, the same value can be set for both.

4.4 Simulation and postprocessing

4.4.1 Simulations

Once all conditions are set, the simulation control files can be written. In this step the case solver is defined, as well as the total computation time, the Δt and other variables such as the start time, the data write interval and other simulation settings.

For this project, four simulations will be performed to evaluate the flow behaviour around the sail, two steady state simulations and two transient state simulations.

The **first simulation** will use the *simpleFoam* application, which is an incompressible steady state solver.

As the flow velocity is way below the 0.2 mach mark, the case can be considered completely incompressible.

For this first simulation, the turbulence properties (annex N) will be set to laminar. The simulation will be set to 600 iterations for the solver to converge to a solution.

The **second simulation** will also use the *simpleFoam* application, this time the simulation type will be set to solve using a RAS model.

When choosing between the three analysed RAS models *kEpsilon*, *kOmegaSST* (two-equation solvers) and *SpallartAllmaras* (one-equation solver), their respective features and behaviour must be taken into account. As explained in Lojek's study[12]:

SpallartAllmaras is classified as more suitable for mildly turbulent external flows, and lacks performance with flows with strong separation.

kEpsilon also performs poorly for complex flows with severe pressure gradient and separation, as well as with high streamline curvature. It is more suitable for initial iterations and screenings of alternative designs.

kOmegaSST is more suitable for complex boundary layers and provides more accurate prediction of flow separation.

As the *kOmegaSST* model should provide more accurate results, it will be used on all the RAS simulations.

Therefore, the second simulation will use the *kOmegaSST* model. This simulation will also be set to 600 iterations.

The **third simulation** performed will use the transient state solver *pimpleFoam* with a laminar model.

The **fourth simulation** will also be performed with *pimpleFoam*, with the turbulent model *kOmegaSST*.

The control dictionaries for each simulation can be seen in order in annexes J, K, L and M.

Note that the number of iterations required ($\frac{endTime}{deltaT}$) is noticeably larger for the transient simulations, as the steady state simulations just need to have enough iterations to converge, but transient simulations require a *deltaT* small enough to maintain the Courant number (CFL) low and stable (simulations crash if CFL grows to infinity) while the *endTime* must be high enough to reach the steady state and instability effects.

The convergence criteria and numerical solvers for each variable can be seen in the *fvSchemes* and *fvSolution* annexes (O, P, Q and R).

4.4.2 Postprocessing

In order to analyse the results, the software *ParaView* will be used. This software uses the meshes and the results from *OpenFoam* cases and generates the three-dimentional geometries with all necessary fields and filters to observe the results.

The main postprocessing utilities will be used to obtain the following results:

- Velocity field
- Velocity vector field
- Streamlines
- Vorticity
- Pressure field
- Cp calculation
- Total drag
- Cd calculation

These main postprocessing utilities are obtained following the steps below (Some of the procedures are explained on Puig's work[16] as well as the ParaView tutorial from the SimScale GmbH channel[8])

The first step will be to extract the fluid and sail blocks as separate filters in order to continue with all the necessary steps.

For the velocity and pressure fields as well as the streamlines, the fluid block is *cut* with the required views. Then, velocity is visualized as a surface finish of U, pressure is visualized as a surface finish of P and streamlines are visualized as a surfaceLIC finish of U.

The streamlines and pressure field on the sail can also be visualized in 3D applying the *streamlines* filter on the fluid and the surface finish of P on the sail block, respectively.

In order to visualize the velocity vector field, the fluid block is *sectioned* at the desired view, and the *glyph* filter is applied, oriented with the U field. The result is a field of scaled vectors that display the direction of the velocity at every point of the domain.

To visualize the vorticity of the velocity, the *gradient of unstructured dataset* filter is applied to the fluid block, with the vorticity option selected. Then the fluid block is *cut* at the desired location and the surface finish of vorticity is selected.

To calculate the total drag, the filter *generate surface normals* is applied to the sail block, then the *calculator* filter is concatenated, computing pressure times normals of x ($p * Normals_X$). Finally, the *integrate variables* filter is applied and the total drag is displayed.

The Cp calculation follows by applying the *calculator* filter to the sail block to calculate Cp $\left(Cp = \frac{p-p_0}{0.5\rho U^2}\right)$. Then, a *slice* of the sail is applied on the desired location (in this case, a slice normal to the Z axis through the center of the sail), and applying the *calculator* filter once again to compute the coords of the slice. Finally, the filter *plot on sorted lines* is applied and the Cp curves are selected with the Y coordinates.

The Cd calculation will be performed outside of *ParaView*, as this software allows only for the analysis of individual timesteps. In order to obtain the drag coefficient for all timesteps, a function is implemented inside the control dictionaries (*controlDict*) of the transient simulations (annexes L and M). The coded

function uses OpenFoam libraries to calculate the aerodynamic coefficients while running the case, therefore the values are computed for all timesteps. Once the case run is finalized, the *gnuplot* software is used to plot the data provided by the implemented Cd function. To display the plot, go to the folder 0 inside the postaProcessing folder of the case with the terminal, write gnuplot to open the ploting utility and finally type *plot './forceCoeffs.dat' u 1:3 w l* to generate the plot.

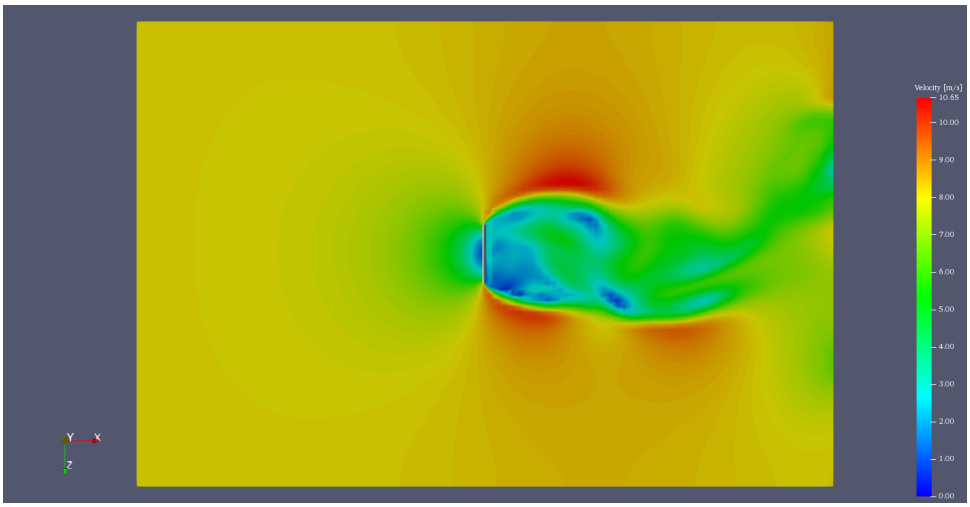
After applying all utilities described, the results can be seen on the results section (4.5).

4.5 Results

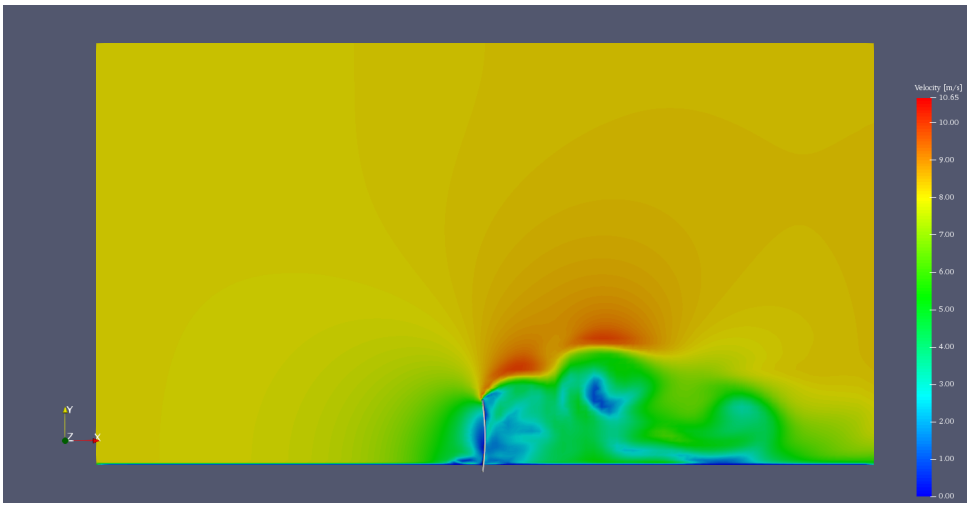
4.5.1 Velocity fields

The velocity fields from each simulation, seen as sections normal to the y and z axis respectively and centered to the sail's center, result as following:

Steady state laminar simulation:



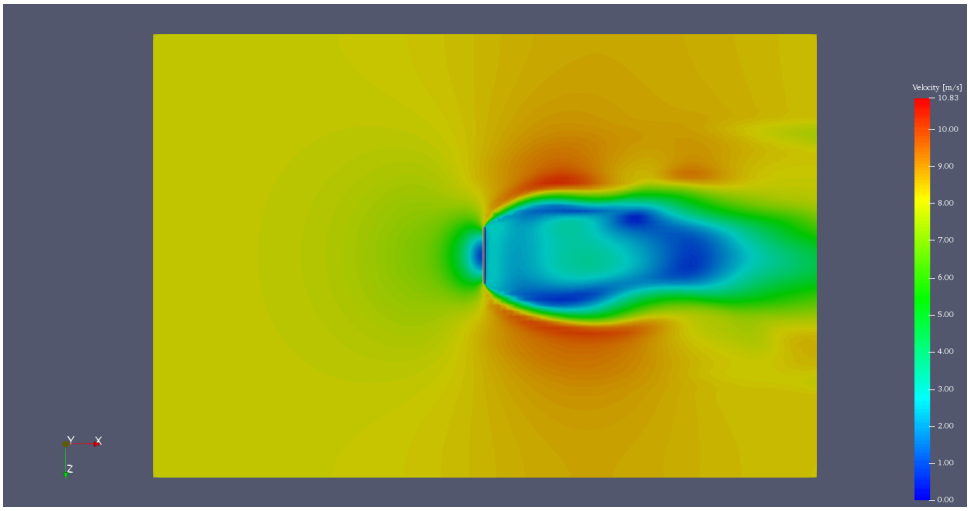
(a) Top view.



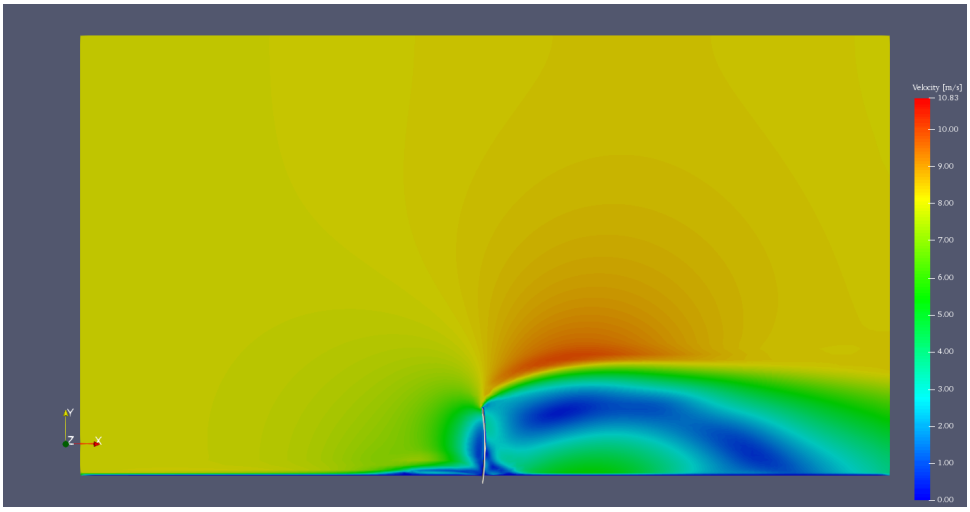
(b) Side view.

Figure 4.5.1: Velocity field of the steady state laminar simulation. [m/s]

Steady state RAS simulation:



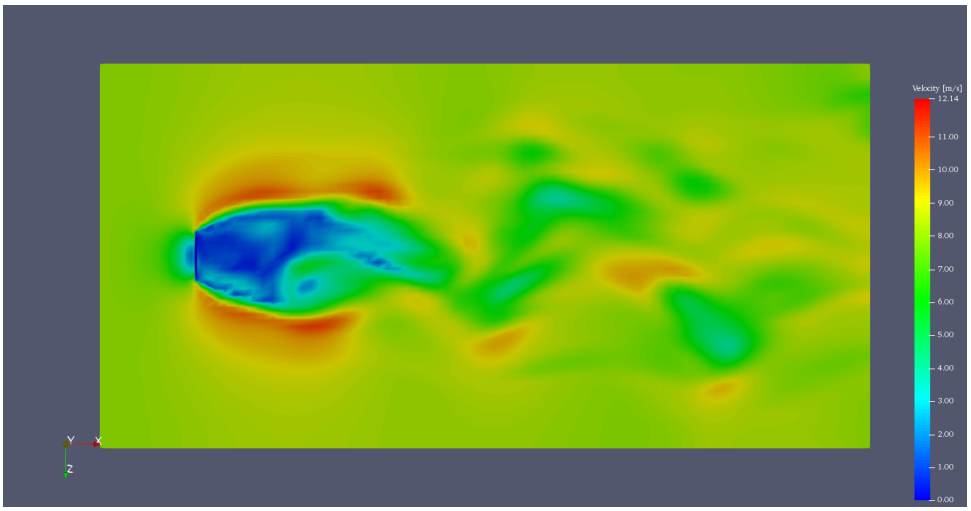
(a) Top view.



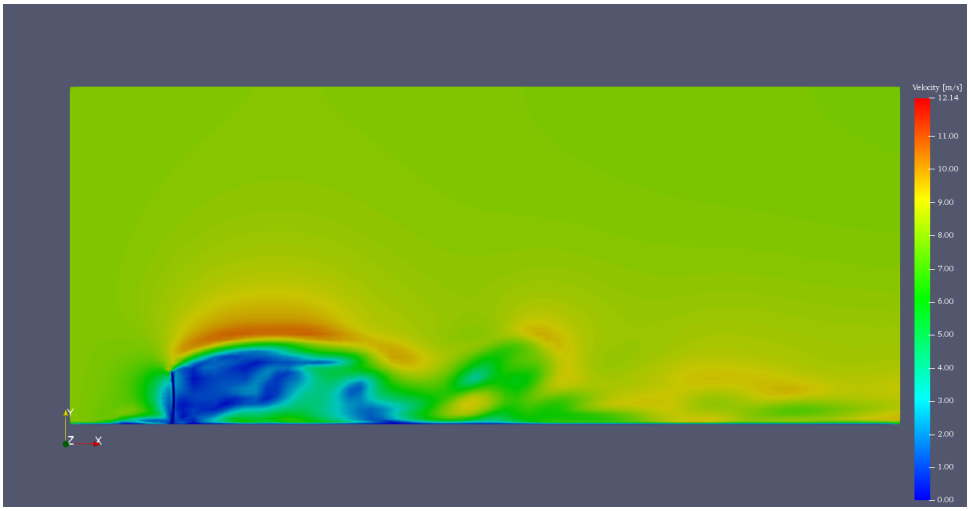
(b) Side view.

Figure 4.5.2: Velocity field of the steady state RAS simulation. [m/s]

Transient state laminar simulation:



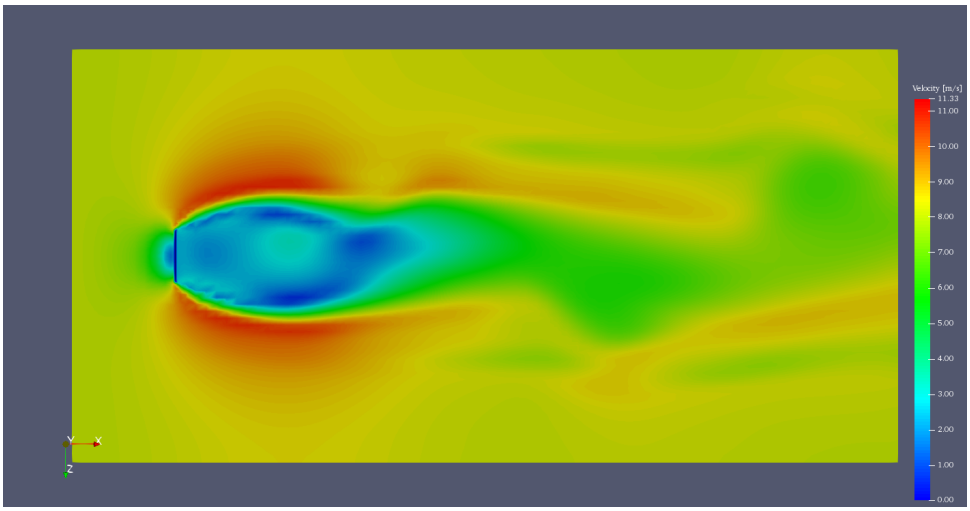
(a) Top view.



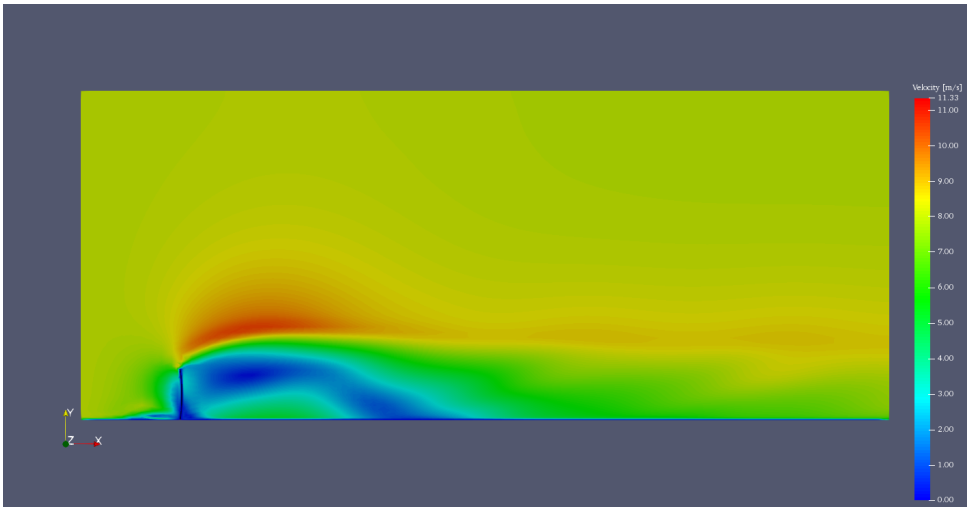
(b) Side view.

Figure 4.5.3: Velocity field of the transient state laminar simulation. [m/s]

Transient state RAS simulation:



(a) Top view.

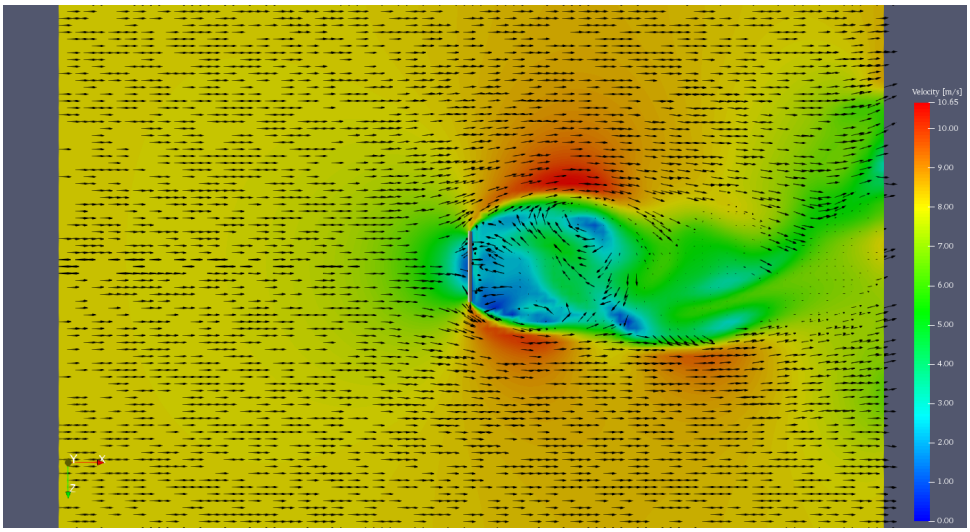


(b) Side view.

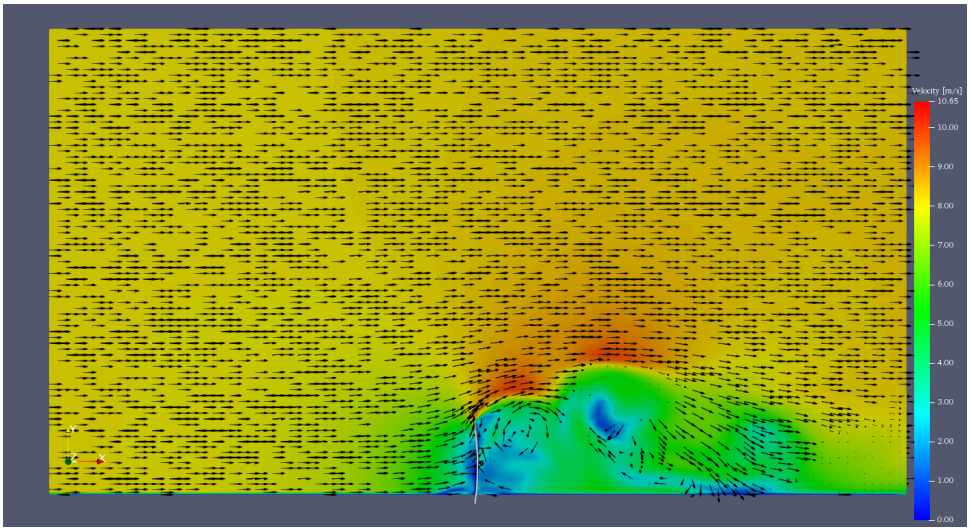
Figure 4.5.4: Velocity field of the transient state RAS simulation. [m/s]

4.5.2 Velocity vector fields

Steady state laminar simulation:



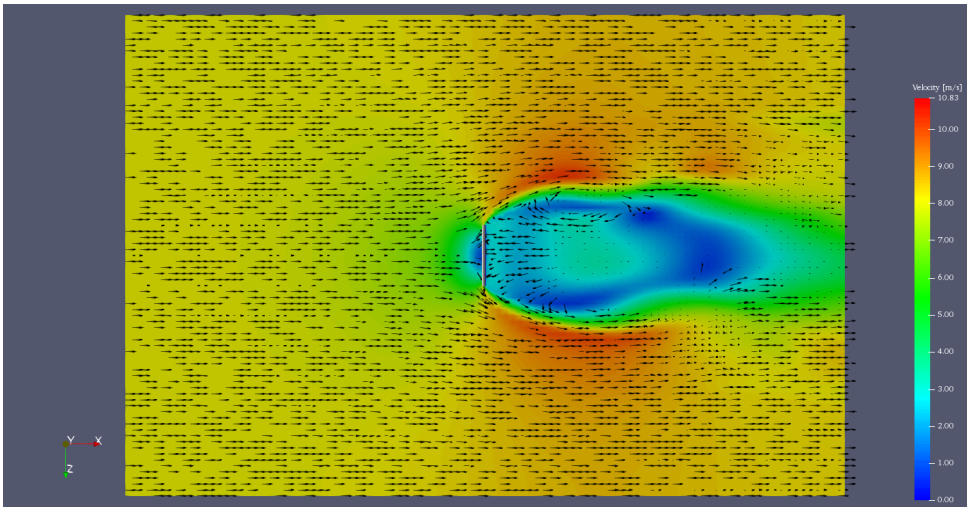
(a) Top view.



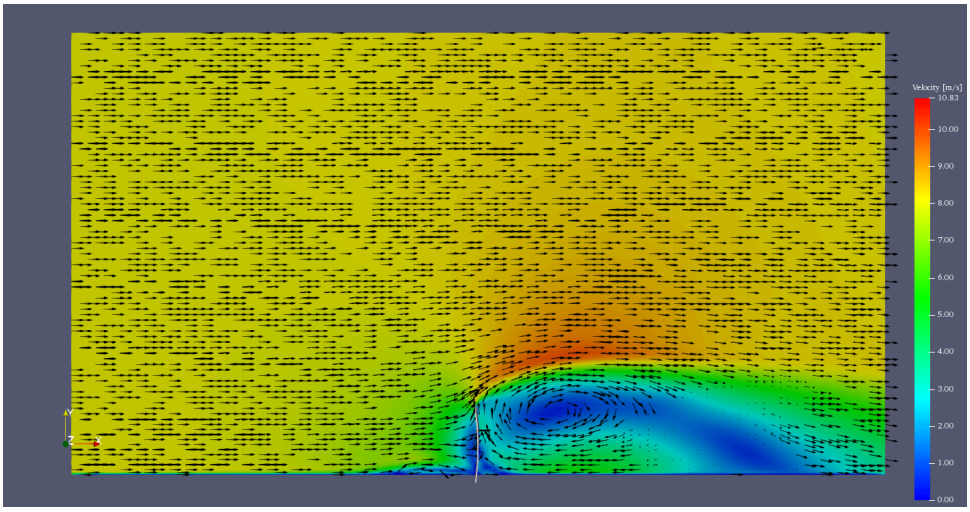
(b) Side view.

Figure 4.5.5: Velocity vector field of the steady state laminar simulation. [m/s]

Steady state RAS simulation:



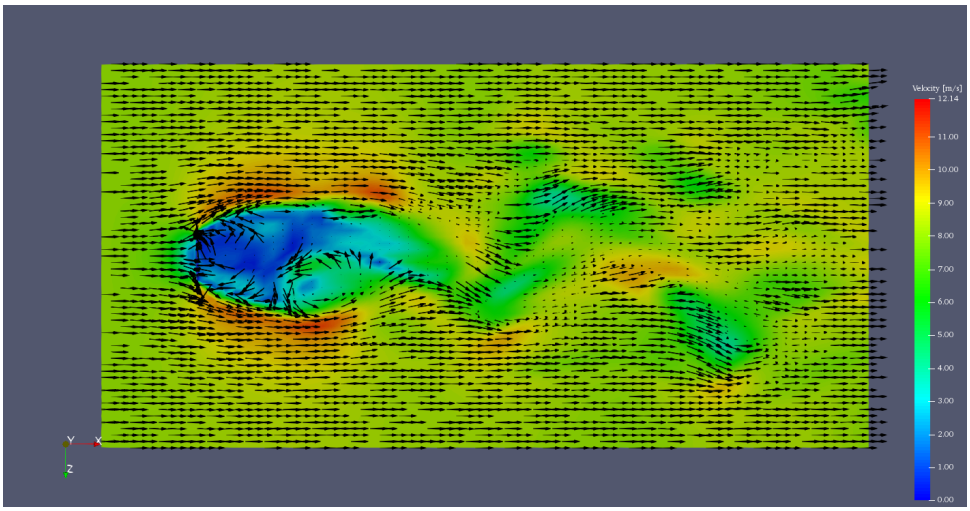
(a) Top view.



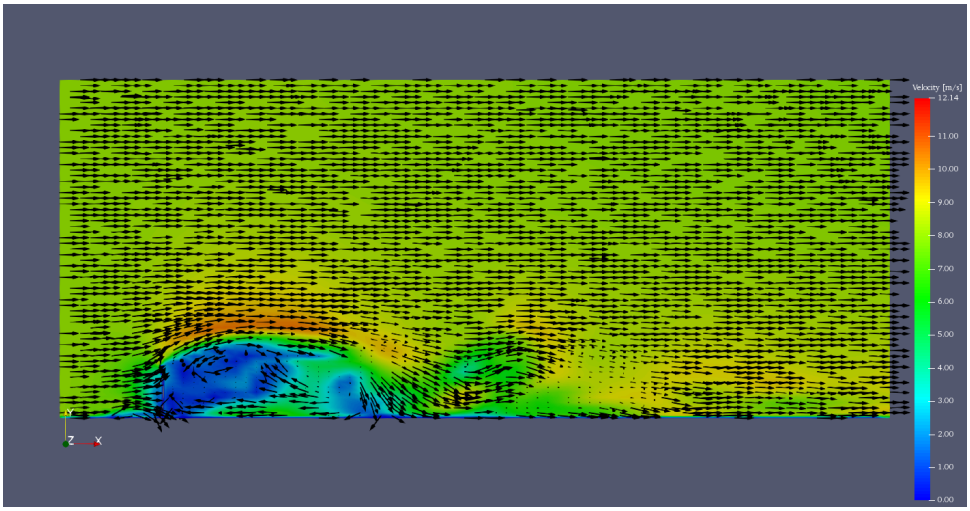
(b) Side view.

Figure 4.5.6: Velocity vector field of the steady state RAS simulation. [m/s]

Transient state laminar simulation:



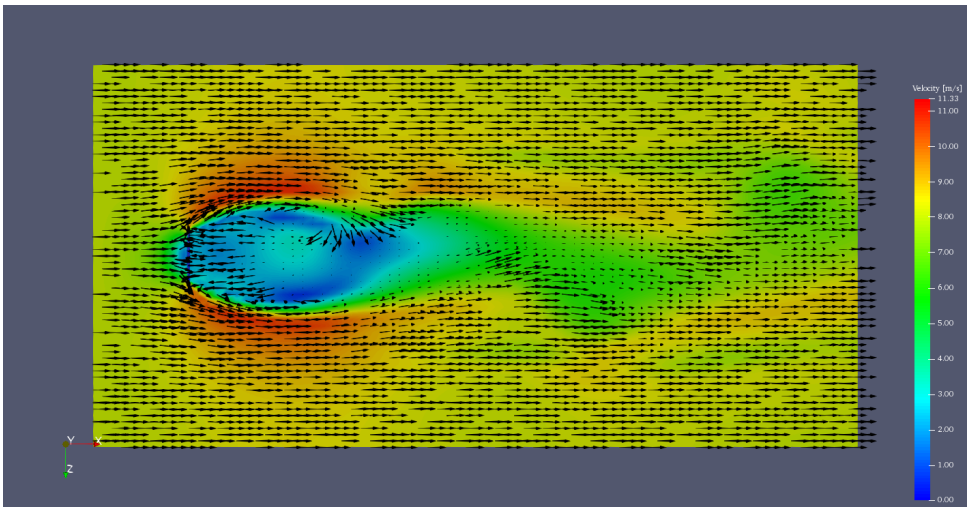
(a) Top view.



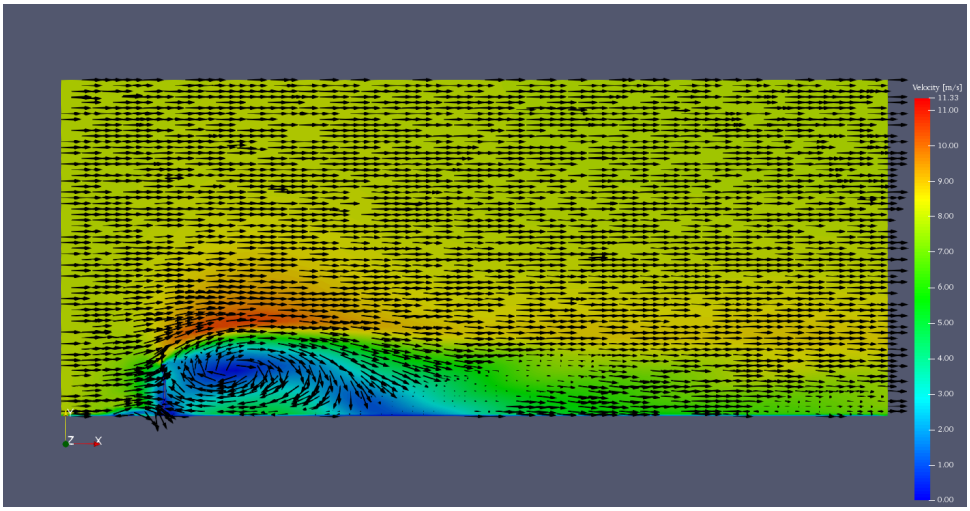
(b) Side view.

Figure 4.5.7: Velocity vector field of the transient state laminar simulation. [m/s]

Transient state RAS simulation:



(a) Top view.

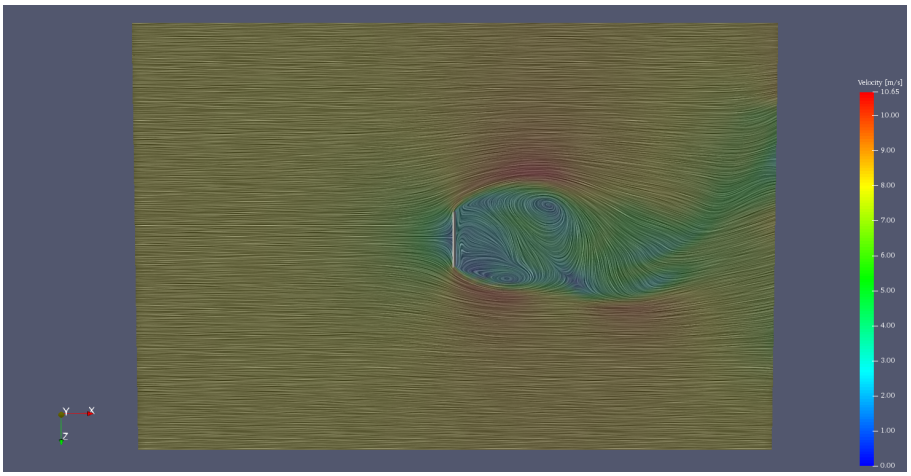


(b) Side view.

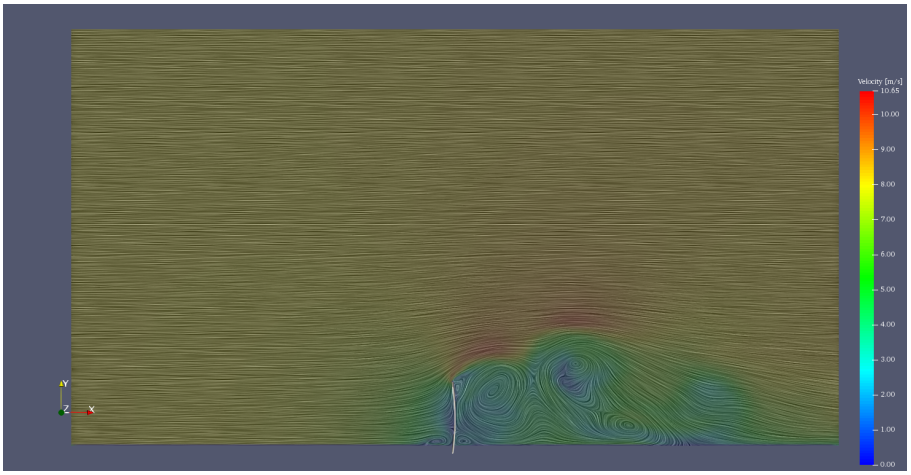
Figure 4.5.8: Velocity vector field of the transient state RAS simulation. [m/s]

4.5.3 Streamlines

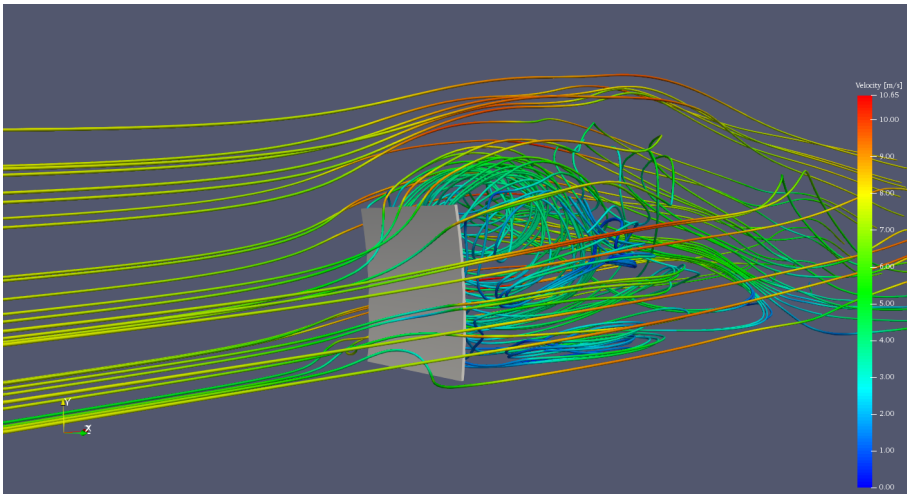
Steady state laminar simulation:



(a) Top view.



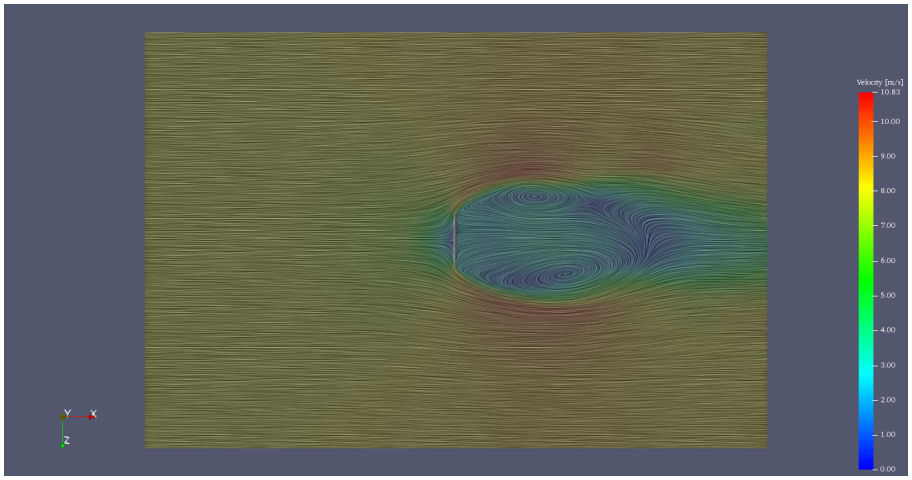
(b) Side view.



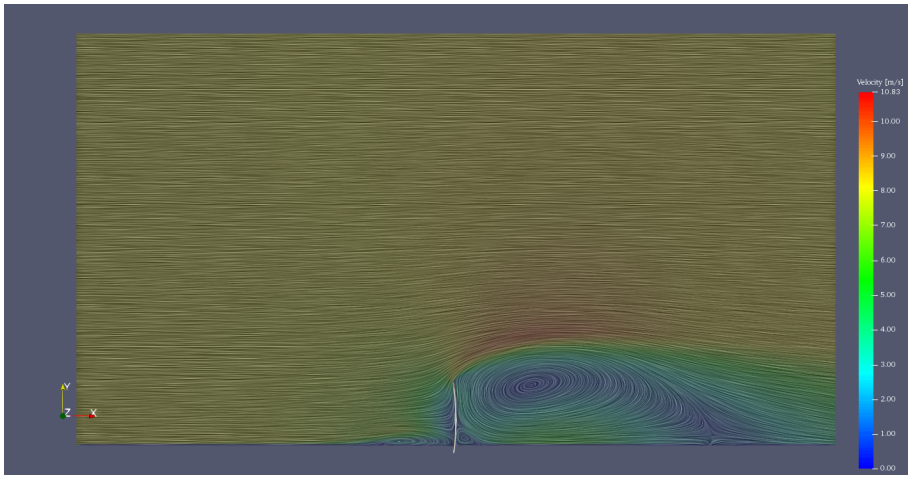
(c) 3D view.

Figure 4.5.9: Streamlines of the steady state laminar simulation.

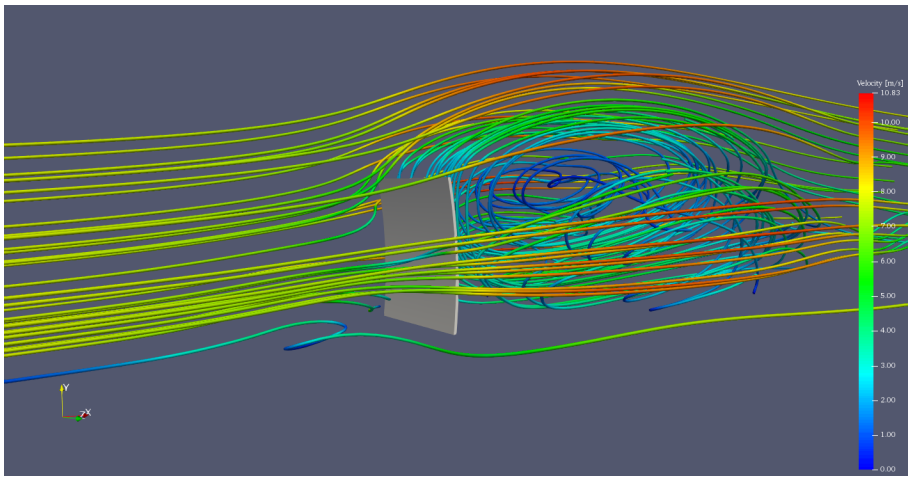
Steady state RAS simulation:



(a) Top view.



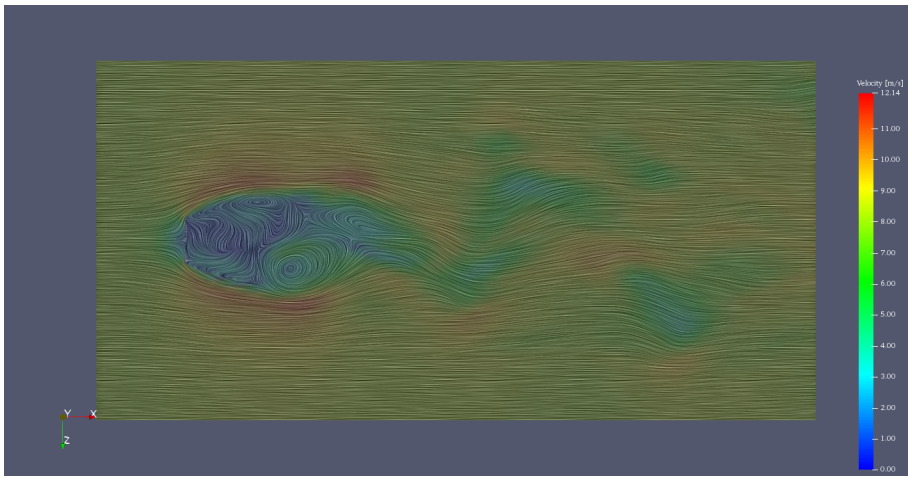
(b) Side view.



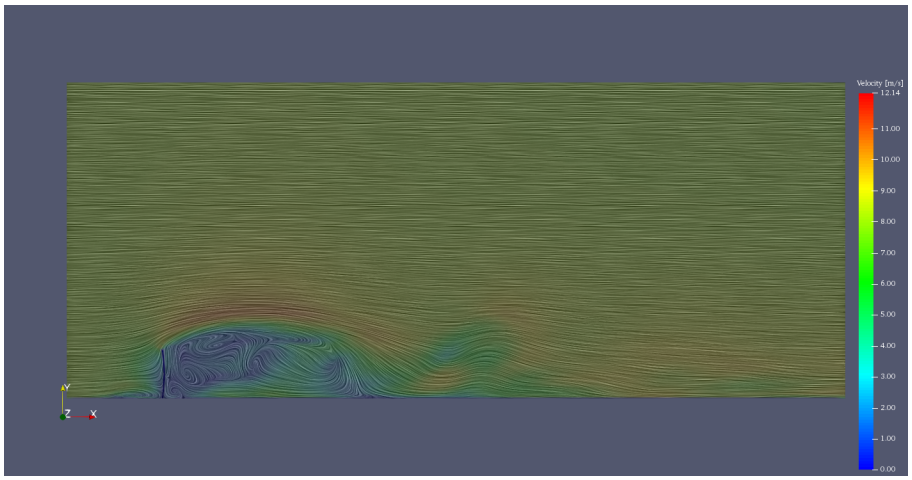
(c) 3D view.

Figure 4.5.10: Streamlines of the steady state RAS simulation.

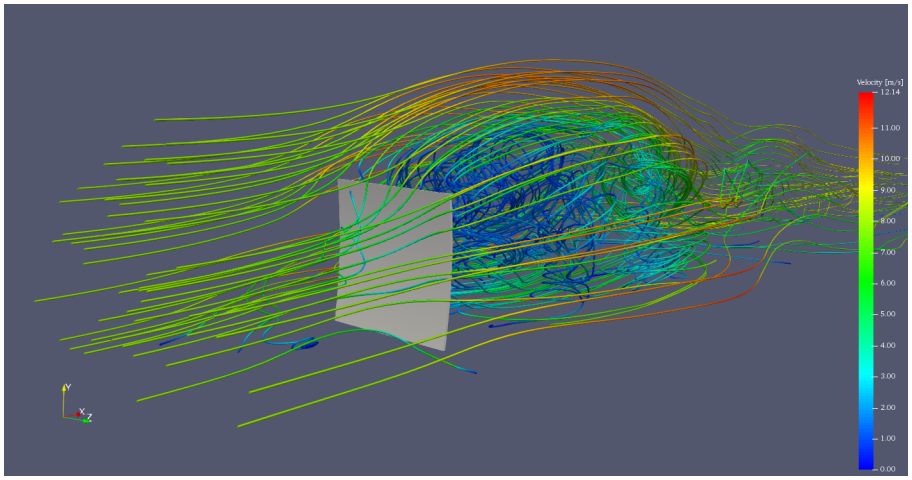
Transient state laminar simulation:



(a) Top view.



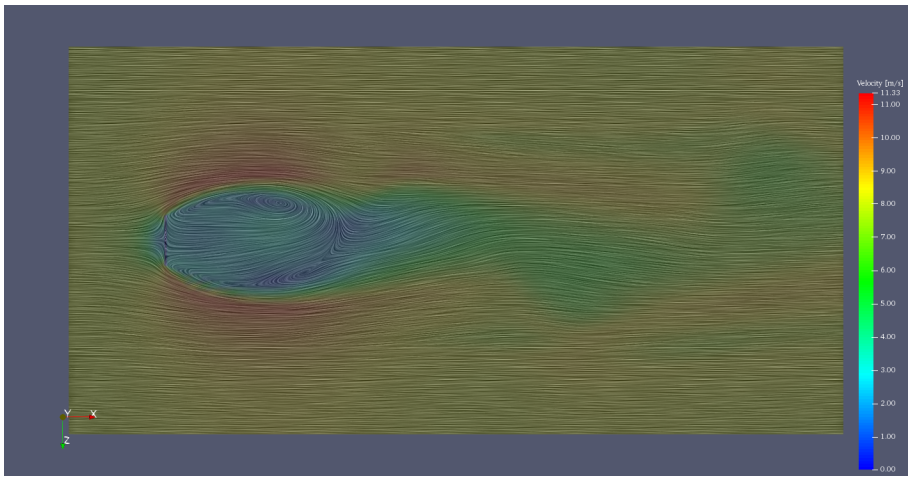
(b) Side view.



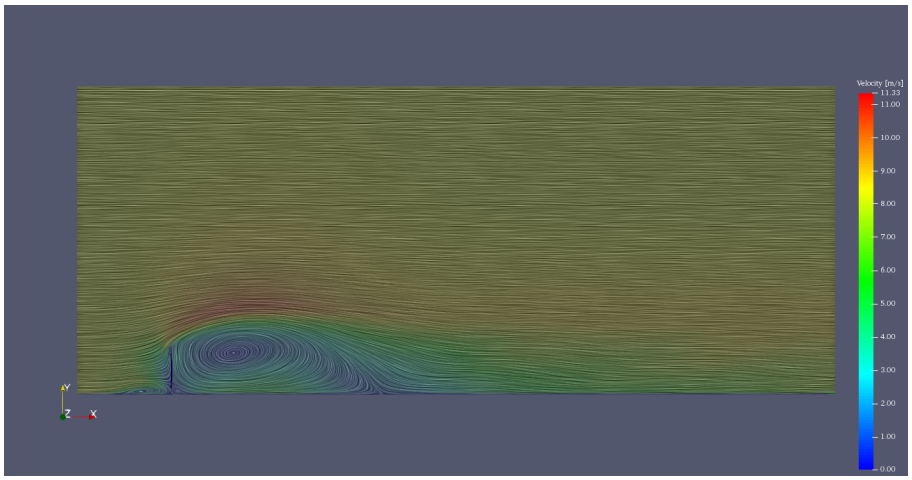
(c) 3D view.

Figure 4.5.11: Streamlines of the transient state laminar simulation.

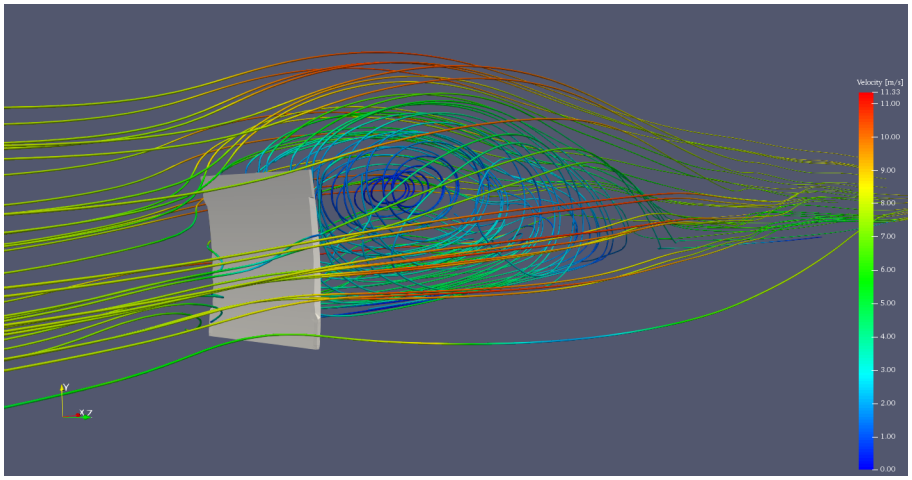
Transient state RAS simulation:



(a) Top view.



(b) Side view.



(c) 3D view.

Figure 4.5.12: Streamlines of the transient state RAS simulation.

4.5.4 Vorticity of the velocity

Transient state laminar simulation:

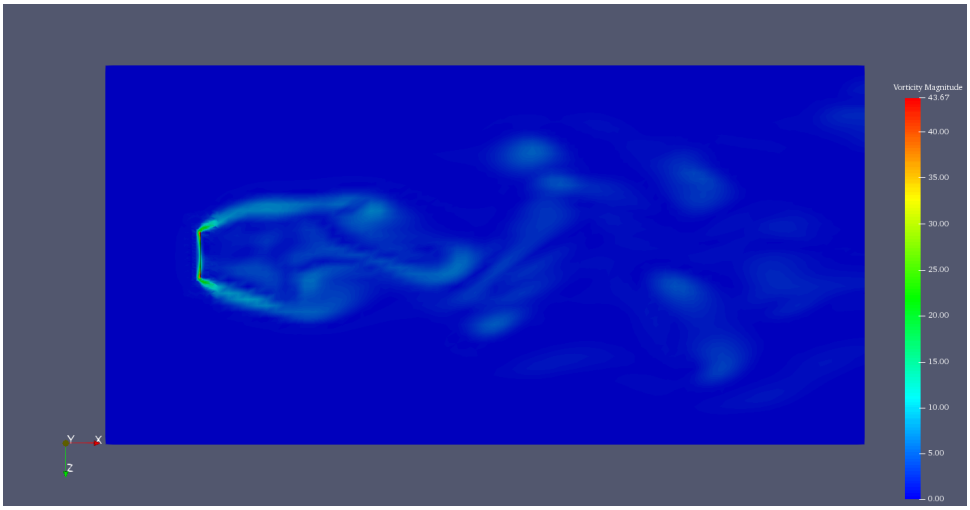


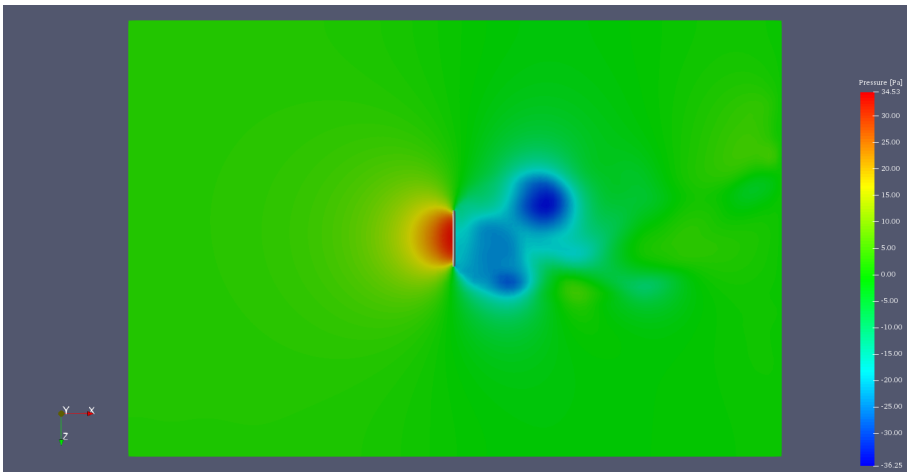
Figure 4.5.13: Vorticity of the velocity field of the transient state laminar simulation (top view).

Transient state RAS simulation:

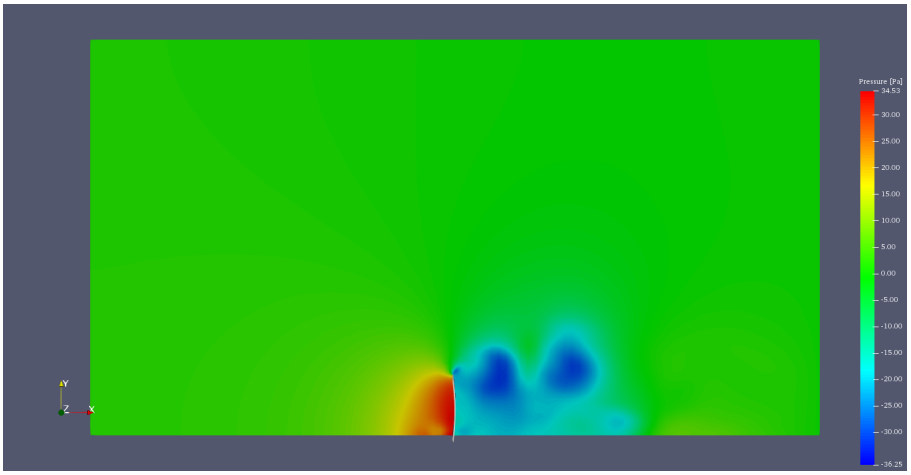
Figure 4.5.14: Vorticity of the velocity field of the transient state RAS simulation (top view).

4.5.5 Pressure fields

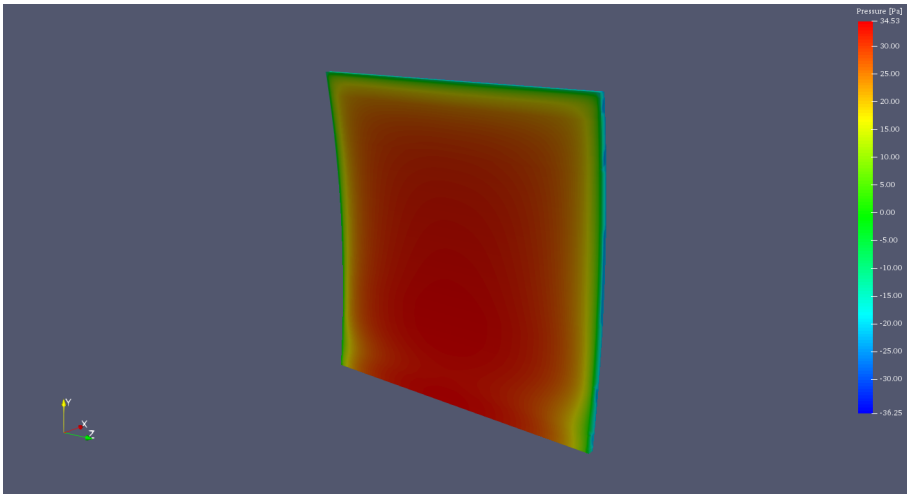
Steady state laminar simulation:



(a) Top view.



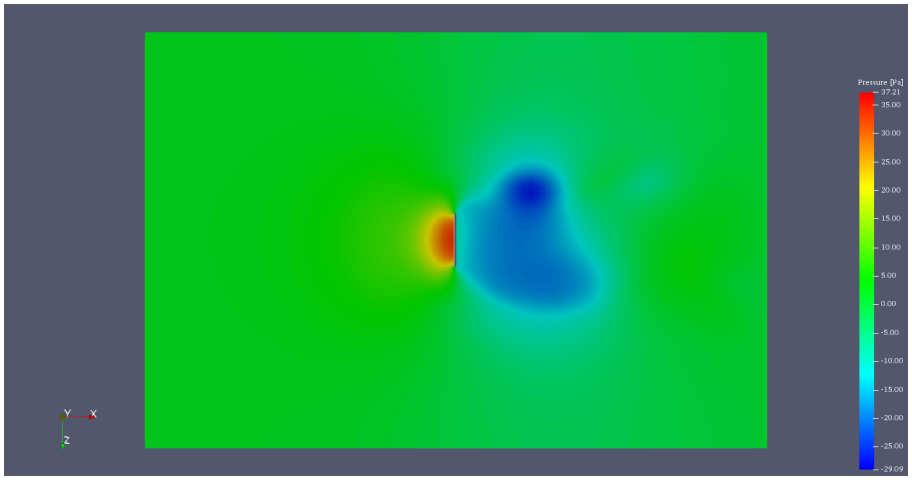
(b) Side view.



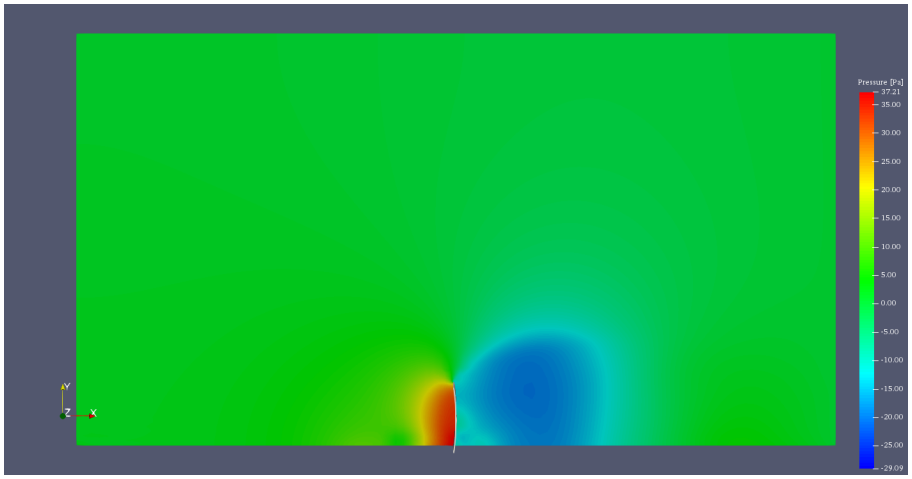
(c) 3D view.

Figure 4.5.15: Pressure field of the steady state laminar simulation. [Pa]

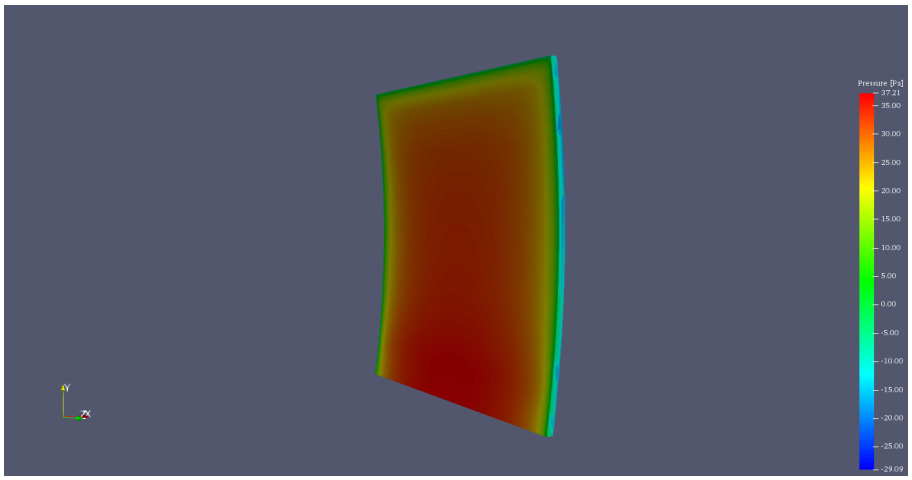
Steady state RAS simulation:



(a) Top view.



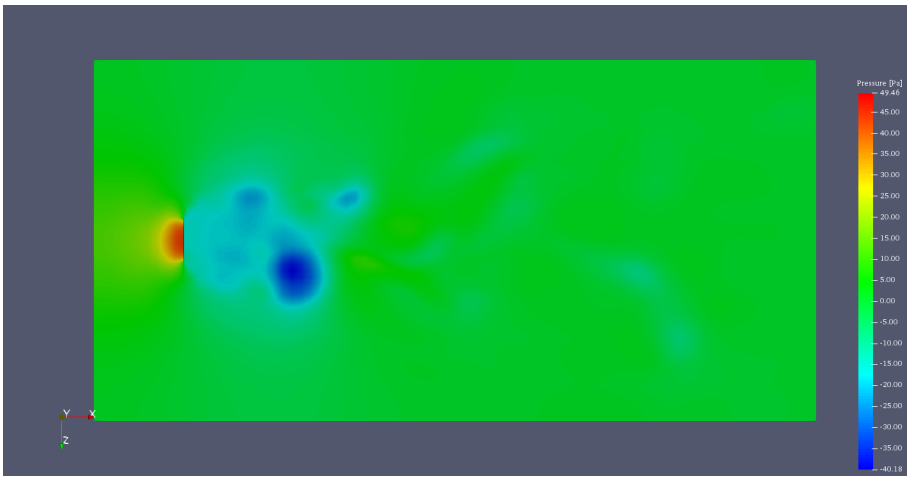
(b) Side view.



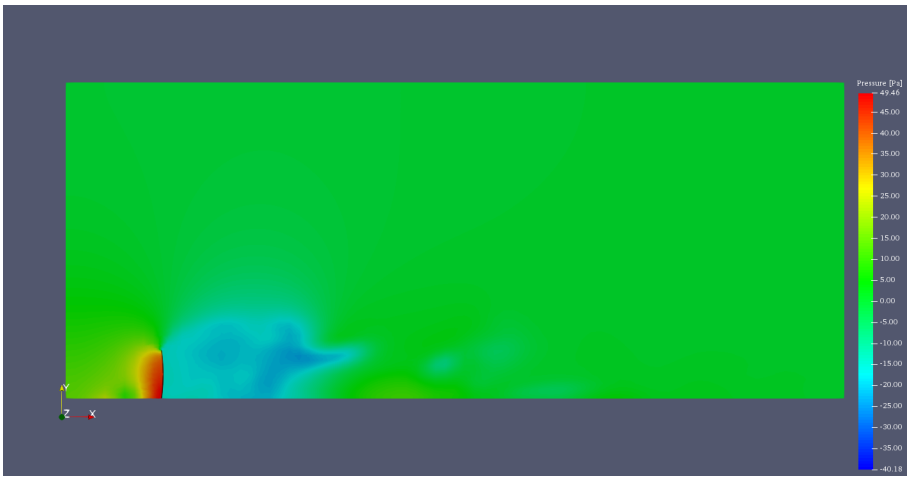
(c) 3D view.

Figure 4.5.16: Pressure field of the steady state RAS simulation. [Pa]

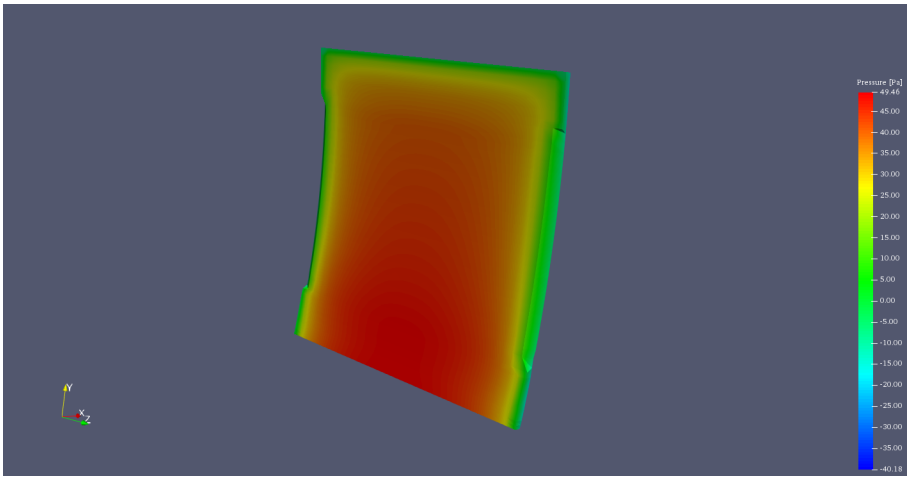
Transient state laminar simulation:



(a) Top view.



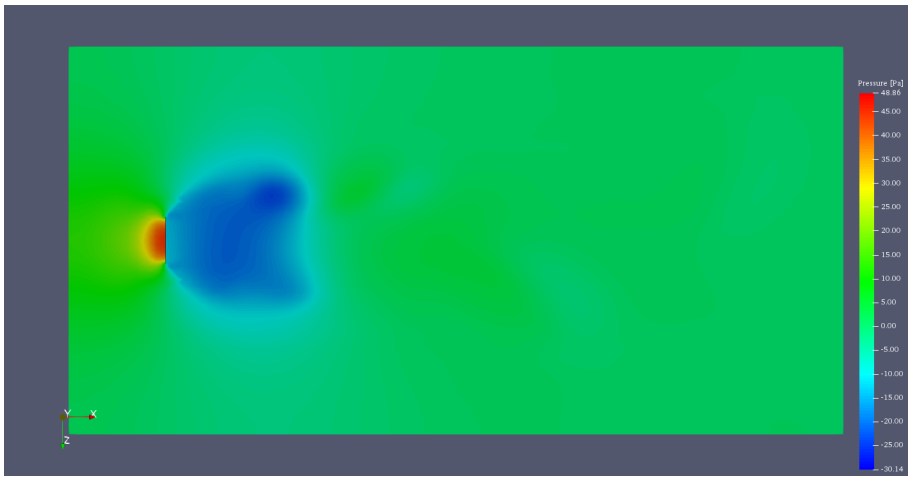
(b) Side view.



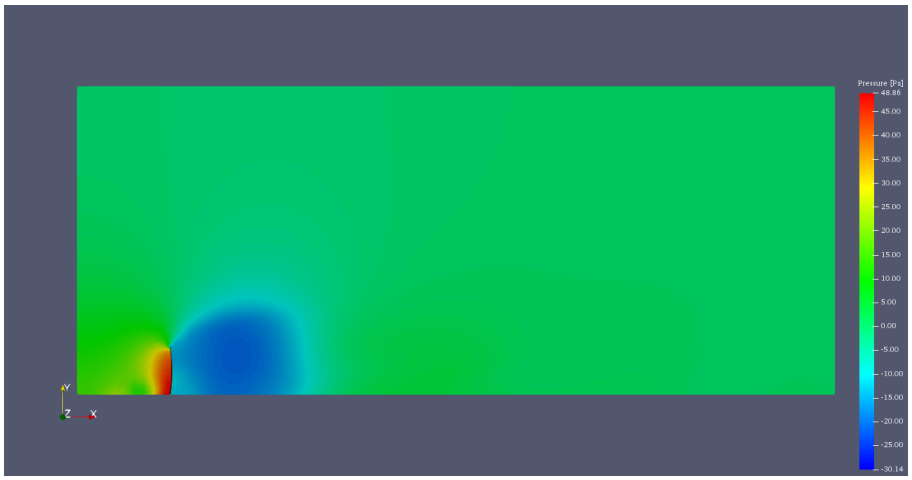
(c) 3D view.

Figure 4.5.17: Pressure field of the transient state laminar simulation. [Pa]

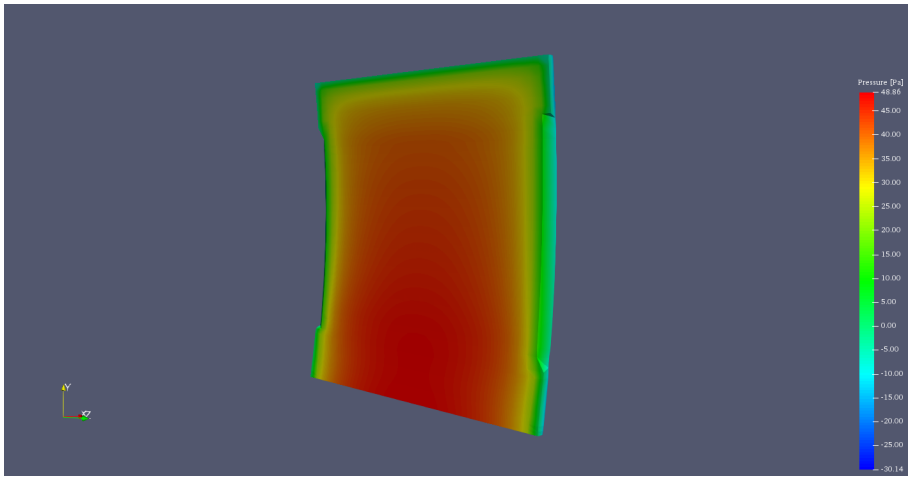
Transient state RAS simulation:



(a) Top view.



(b) Side view.



(c) 3D view.

Figure 4.5.18: Pressure field of the transient state RAS simulation. [Pa]

4.5.6 Pressure coefficient of the sail

Steady state laminar simulation:

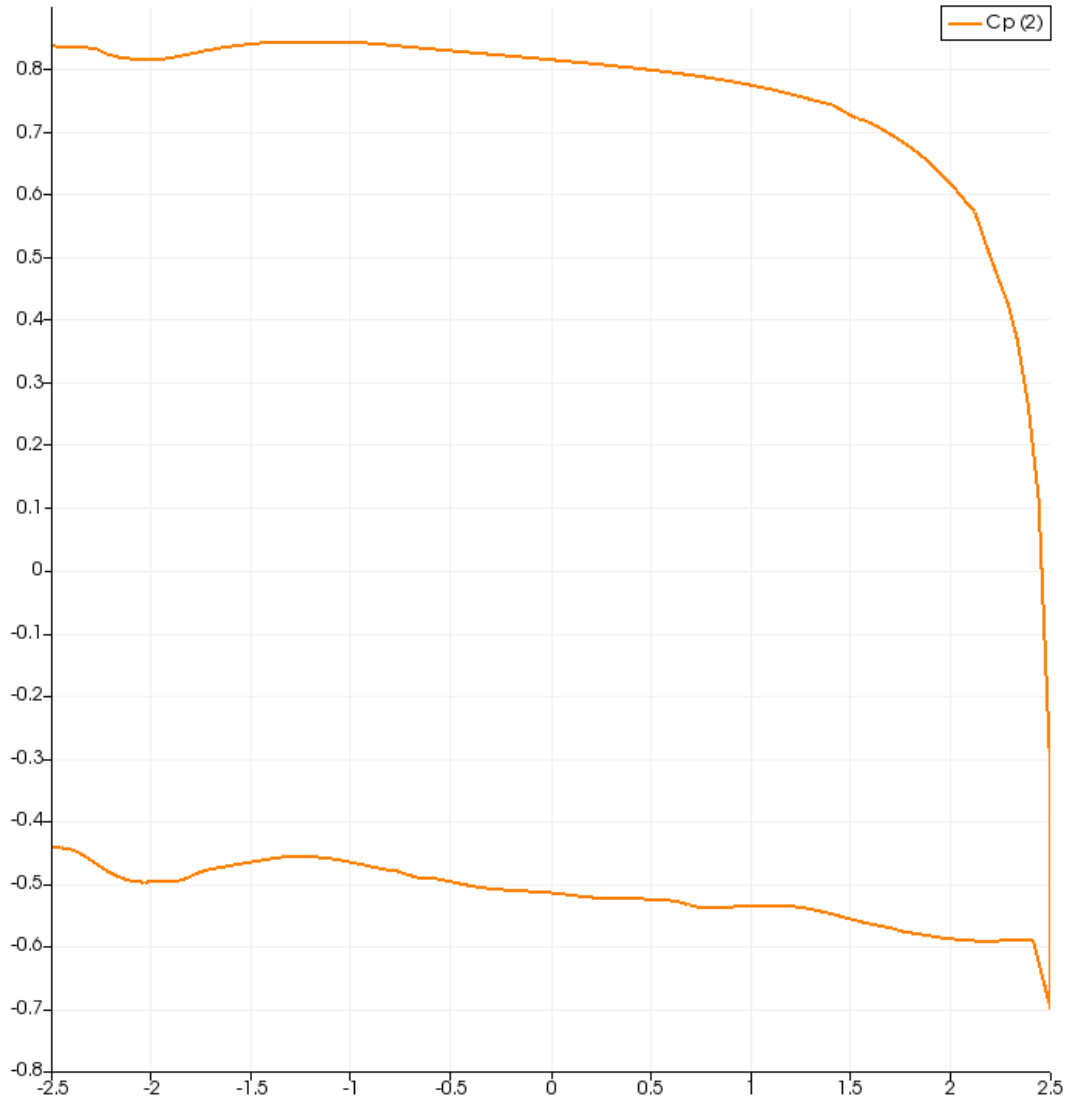


Figure 4.5.19: Pressure coefficient along the middle section of the sail for the steady state laminar simulation.

Steady state RAS simulation:

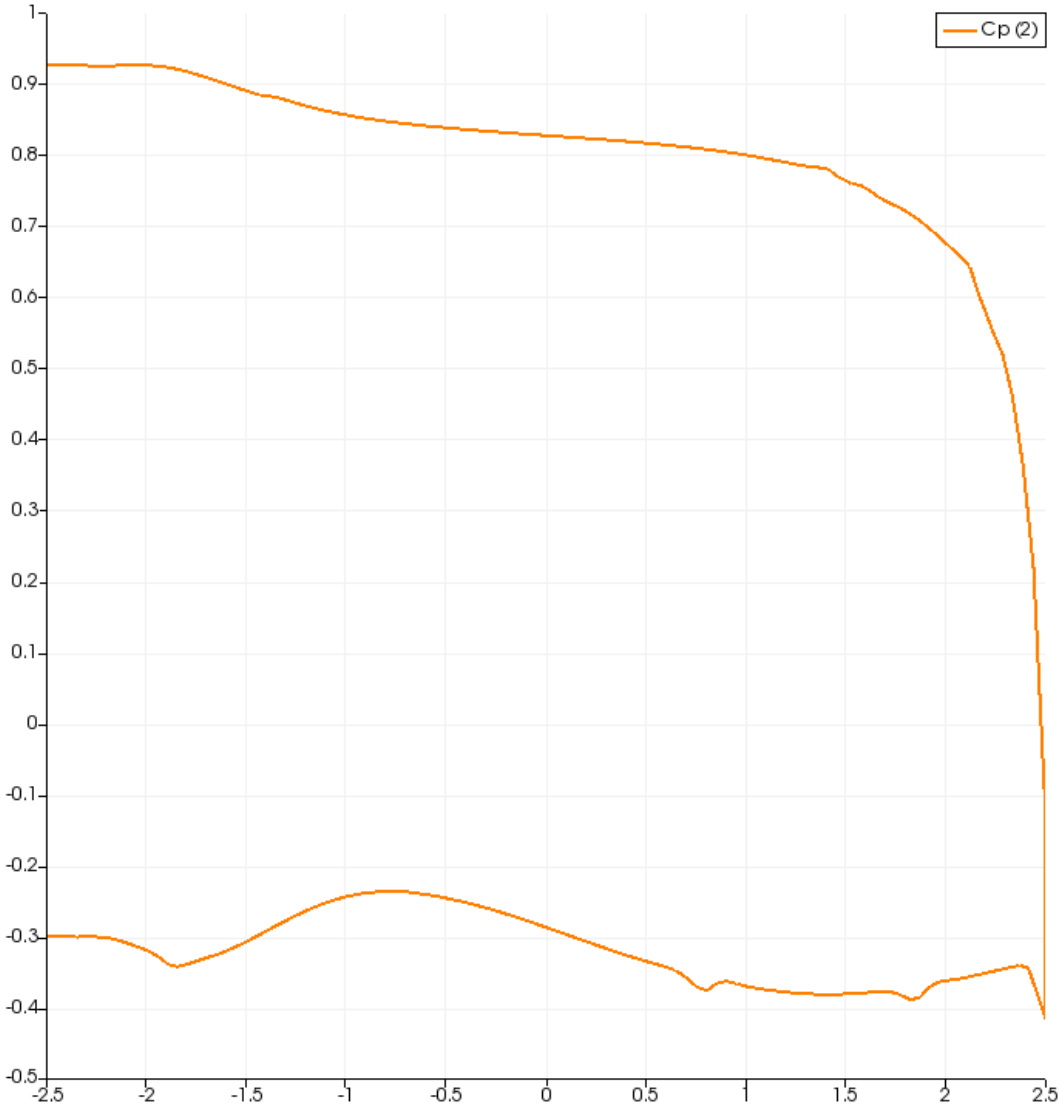


Figure 4.5.20: Pressure coefficient along the middle section of the sail for the steady state RAS simulation.

Transient state laminar simulation:

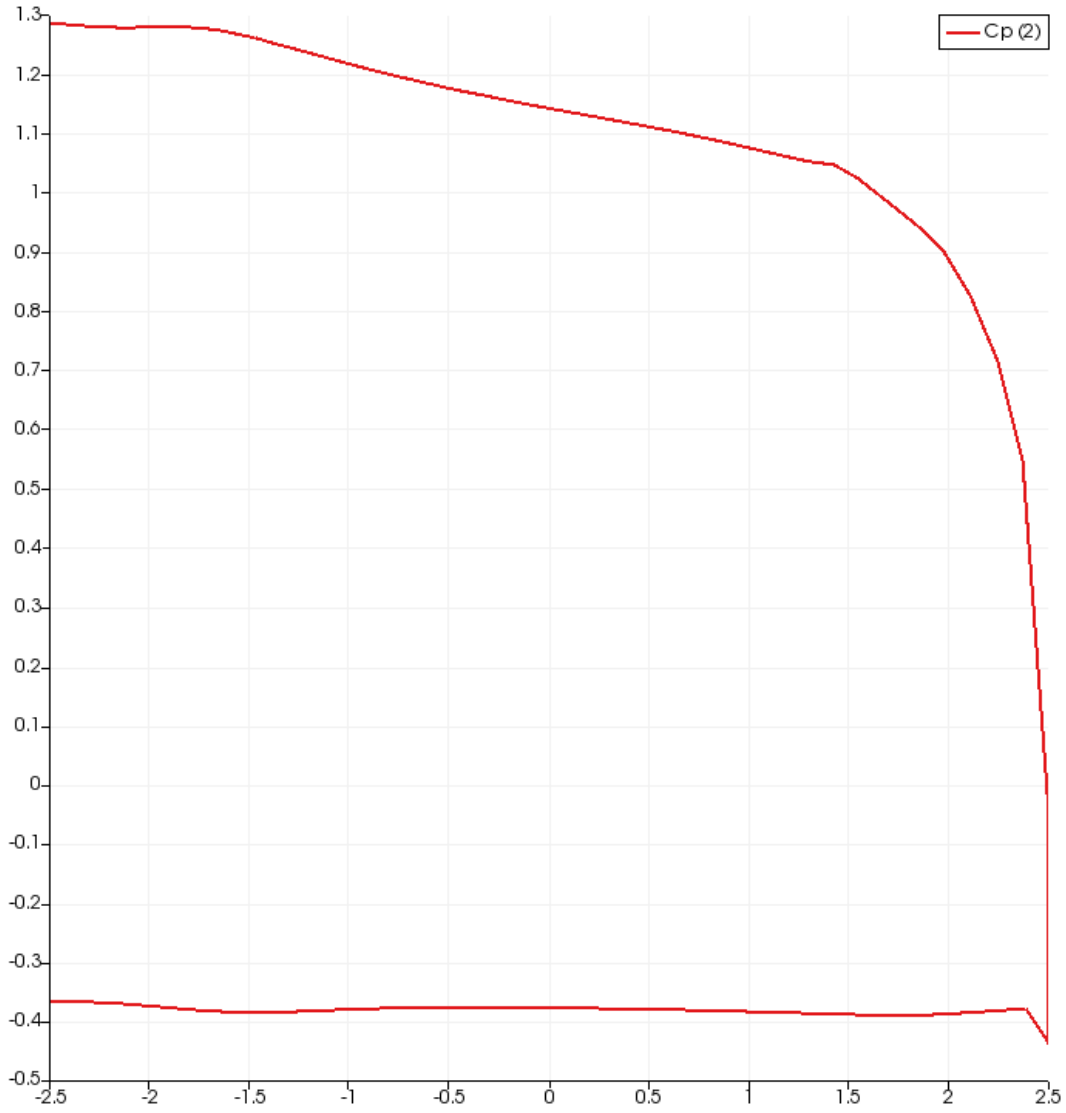


Figure 4.5.21: Pressure coefficient along the middle section of the sail for the transient state laminar simulation.

Transient state RAS simulation:

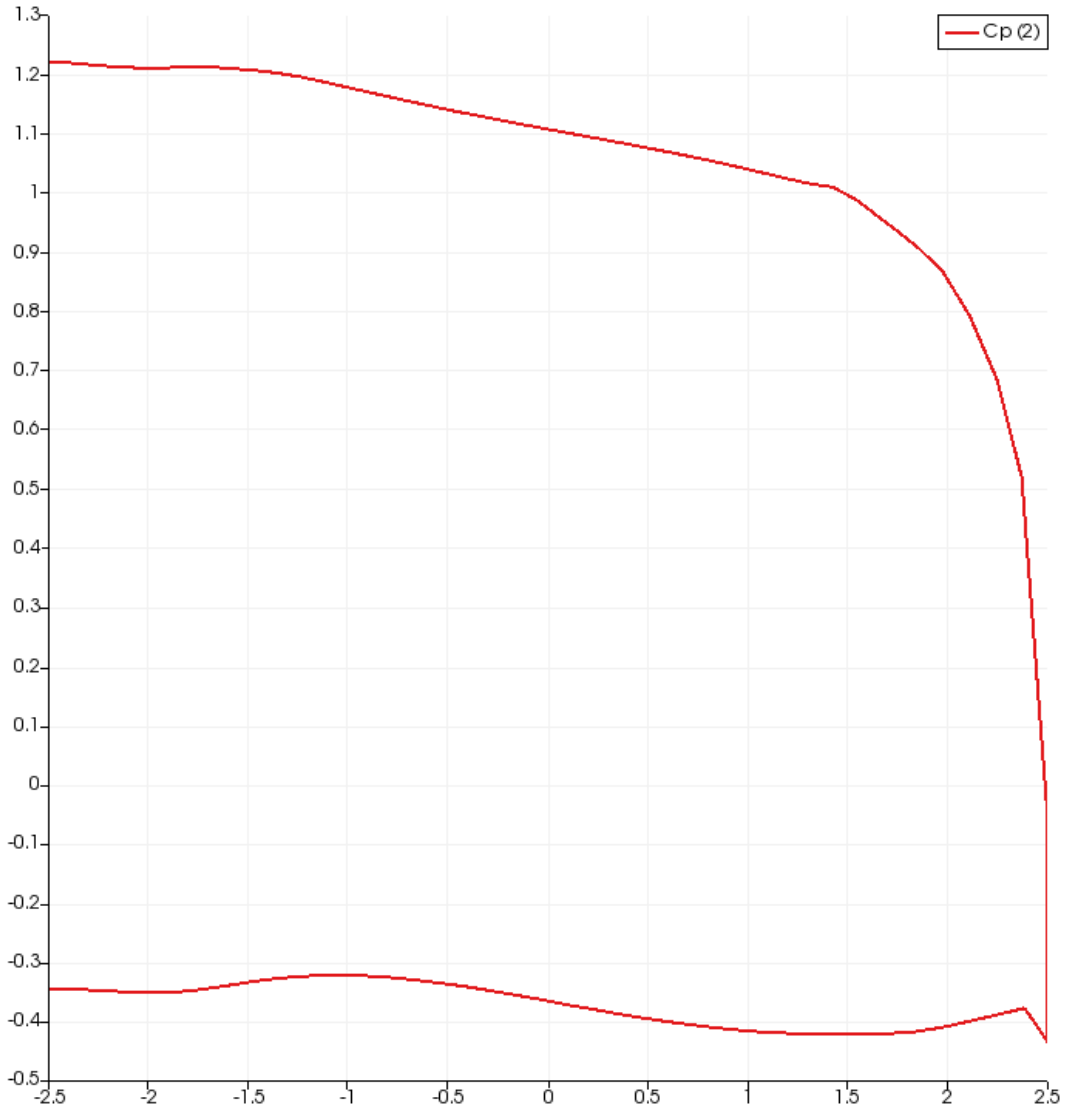


Figure 4.5.22: Pressure coefficient along the middle section of the sail for the transient state RAS simulation.

4.5.7 Total drag calculation

Steady state laminar simulation:

$D = 1120.04N$

Steady state RAS simulation:

$D = 995.22N$

Transient state laminar simulation:

$D = 1256.87N$
At $t = 75s$

Transient state RAS simulation:

$D = 1223.29N$
At $t = 200s$

4.5.8 Drag coefficient plots over time

Transient state laminar simulation:

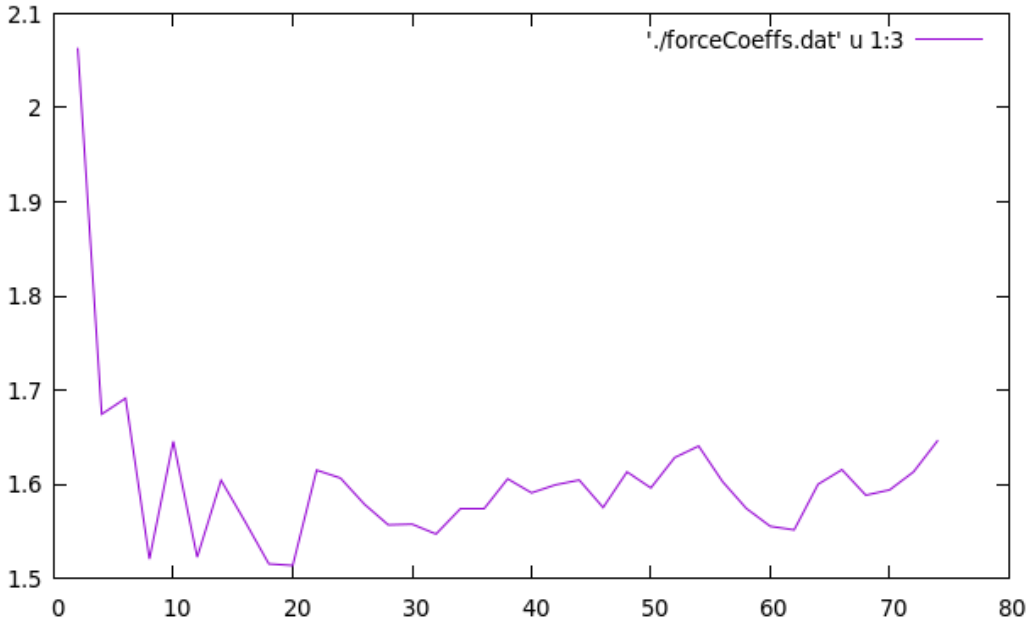


Figure 4.5.23: Total drag coefficient over time for the transient state laminar simulation.

Transient state RAS simulation:

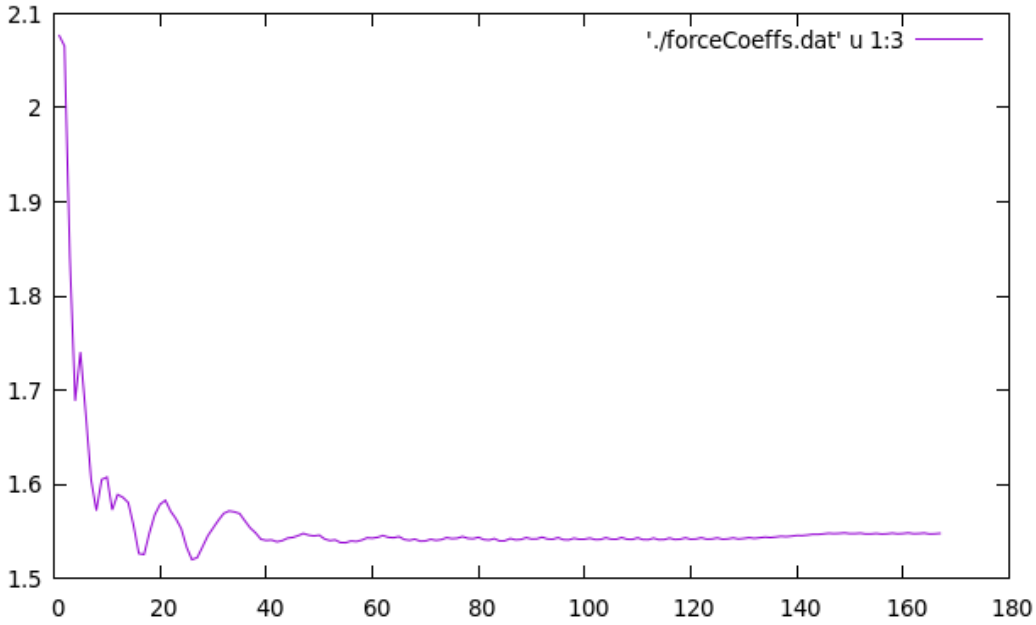


Figure 4.5.24: Total drag coefficient over time for the transient state RAS simulation.

5 Results discussion and conclusions

5.1 Discussion of the results

Once all the results are presented, one can observe that they are very similar for all four simulations, the general behaviour of the fluid is equivalent given the same boundary conditions, which is a clear indicator that all boundary conditions are set correctly and the simulations have worked without any errors.

There can be seen, however, some distinct differences, especially between laminar and RAS simulations.

In order to analyse the results, when studying an aerodynamic problem, the first thing to do is calculate the Reynold's number (Re) which provides key information about the flow characteristics.

$$Re = \frac{UL}{\nu} = \frac{8 \cdot 5}{1.5 \cdot 10^{-5}} = 2.66 \cdot 10^{6} \tag{42}$$

Where U is the freestream velocity of the fluid, L is the characteristic length of the problem (in this case of external flow around a sail, the height of the sail), and ν is the kinematic viscosity of the fluid, in this case air.

The Reynold's number determines if the flow is laminar or turbulent. As seen in figure 5.1.1, the flow regime changes significantly with the Reynold's number.

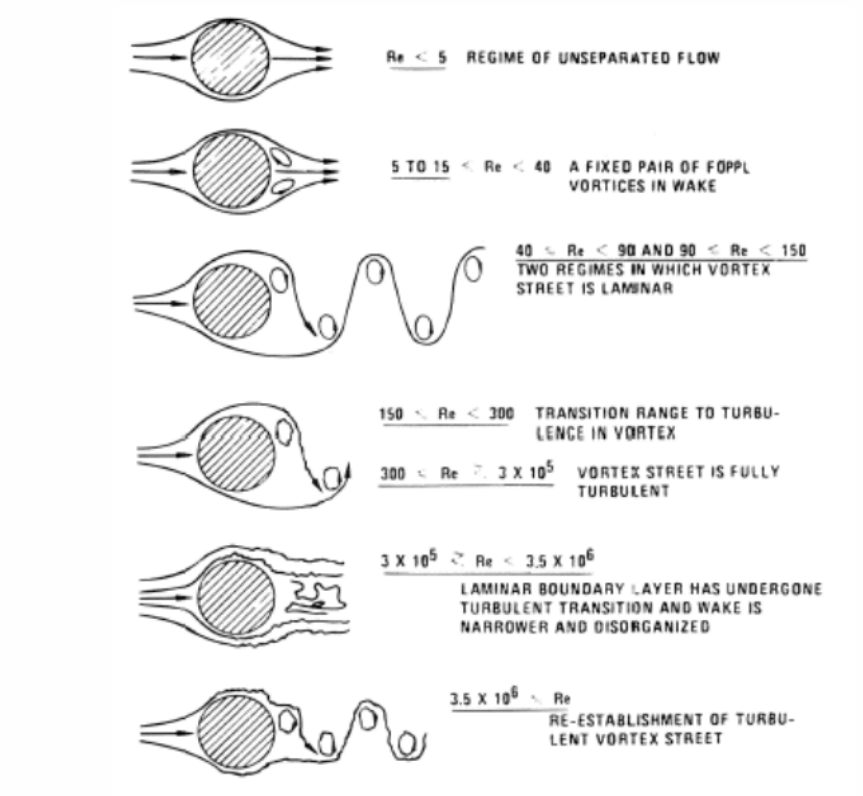


Figure 5.1.1: Fluid flow regimes across a smooth cylinder[2].

There is a critical value of the Reynold's number that determines the transition from laminar flow to turbulent flow. It can be seen in figure 5.1.1 that the vortex transitions to turbulent from $Re = 300$ up to

$Re = 3 \cdot 10^5$. A more concrete value typically used as a critical Re is $Re = 10^5$ [20]. Therefore, the case studied behaves as fully turbulent and the RAS simulations should provide more adequate results.

Comparing the simulation results obtained in this study to the explanation of the wake behaviour with external flow [18], the vortex shedding phenomenon can be seen in all four simulations (see figures in chapters 4.5.1 and 4.5.2).

As the transient simulations include the extended mesh, the wake after the sail can be more accurately analysed.

The steady state RAS simulation clearly shows a double vortex wake, which can also be identified on the transient state RAS simulation, before the wake sheds and forms the Von Kármán vortex street (see RAS simulations on chapter 4.5.3). This Von Kármán vortex street can be observed as the vorticity of the velocity field (see chapter 4.5.4), where it can be seen that the vorticity of the RAS simulation on the vortex street is practically negligible. This indicates that the vortices formed close to the sail at that timestep have low fluid rotation.

The pressure fields (chapter 4.5.5) show the main source of drag for the sail, the pressure drag due to the difference in pressure between the front and the back of the sail. This pressure is not uniform along all the surface of the sail, the distribution can be observed on the Cp graphs (chapter 4.5.6), which are all almost identical and show the pressure is mostly uniform at the center of the sail and decreases getting closer to the edge of the sail. Notice also that, as this simulation has been performed with the sail touching the ground, the Cp at that side of the sail does not decrease closer to the edge, as the fluid has no way under the sail.

Regarding the total drag values (chapter 4.5.7), steady state simulations result in lower drag values, which is a result of using a steady state solver, which does not take into account the time variability.

However, it is also noticeable that both steady and transient state RAS simulations have a slightly lower drag than their laminar counterparts. This is due to the turbulence model better computing the behaviour of the wake with large separation and pressure gradient, but the results do not differ too much.

Finally, the Cd plots over time for both laminar and RAS transient state simulations show the instability of the wake. The laminar simulation has a chaotic behaviour due to its lack of dampening through turbulence, as for the RAS simulation, the turbulence model implements this dampening and the instability becomes regular, generating what is called Vortex-induced vibrations, as the vortex shedding shifts the center of pressure downstream of the sail, therefore the drag force changes direction back and forth following a sinusoidal pattern with a characteristic frequency.

After all the study is completed, the shape of the sail could be recalculated using the data from the pressure distribution along the surface of the sail, which would provide the distribution of drag. These values could be the input in a FEM code to calculate deformation, knowing the mechanical properties of the sail material (Dacron), and recreating the aerodynamic study with the updated geometry.

There is also the possibility of setting a more complex case with a dynamic mesh that deformed as the pressure forces acted upon it. However this would require a solver that can solve both aerodynamic and structural data sets.

Regarding this iteration process to recalculate the shape, in D. Boote and M. Caponetto's work [3] the Vortex Lattice Method is used to code a program to calculate the equilibrium shape.

5.2 Conclusions

After completing the project, I have found that the iteration methods for the shape calculation would require more time than what I had disposal of, especially because I had foreseen that the aerodynamic study would require a lot of computation time and rerunning after correcting variables.

Aside from that, as I was also more interested in the aerodynamic study and the development and running of the CFD simulations, so I decided to avoid coding an iterating software but rather manually calculate a first iteration and define that the final pressure distribution could be implemented to recalculate the shape.

Regarding the timing scheduled on the project charter, after having completed the project I would re-structure all the work hours for each task, mainly focusing the calculation of the shape and especially the simulation and postprocessing. My initial idea was to just perform the steady state simulations, but as I was investigating the utilities for aerodynamic studies I found out that transient state simulations show a more accurate behaviour and especially offer the time variable to observe the diverse flow states and instabilities through time, therefore I decided to implement transient state simulations as well. Transient simulations took 40 hours of computation time, and I had to rerun them multiple times when adjusting coefficients and correcting errors, especially the first runs that crashed a lot, and I did not account for that type of delay. However, i used the computing time to write the report so at least I did not waste time while computing.

Some other problems I had while setting the transient simulations, was that the steady state flow did not fully set until around 100 seconds on the RAS simulation (the laminar simulation showed the steady state and vortex shedding formation in 75 seconds so I did not have much problems there). As I intended to show the vortex street formation, I needed a simulation 200 seconds long, but if the timesteps were too small the simulation would have taken longer than 40 hours. I tried to bypass that by using bigger timesteps, however, if they were too big, the simulation would crash halfway as the Courant number would grow to infinity. By slightly adjusting the turbulence variables and the timesteps finally I reached a stable run and could obtain the transient state RAS simulation results.

I also had a problem with transient simulations as the simulations were coded to write results every 5 timesteps initially, but one run crashed because the computer ran out of memory to write the results. As I had generated a longer mesh to observe the wake behaviour, there were a lot more cells to compute and a lot more data on the results, so I had to generate a new thicker mesh (losing a bit of resolution) and set the simulation to write results less frequently (losing time resolution), but the downgrade in resolution was not so bad that results could not be observed, so I saved the conditions as valid.

It can also be seen in transient simulations that the sail has a bevel on both left and right edges of the sail. This is not how the original geometry of the sail was designed, but rather the result of generating the fluid mesh with *snappyHexMesh* because the mesh had to be made thicker and it didn't properly catch on the sail's edges. However, the deformation is so small that it was considered to have no effect on the results, and if however more precision is desired, the only thing needed is to make the mesh finer and the *snappyHexMesh* will better generate the edge of the sail.

As a final conclusion, I think I solved all the problems I found in the way in a very effective manner, and I am very content with the results, as I have learnt a lot about the shape calculation and the recreation and study of aerodynamic cases. If I had to redo/continue the project with more time, I would focus on performing the second iteration of the shape calculation and using a different solver to perform the CFD simulation and investigate what is the best solver and turbulence model to study this kind of external flow.

References

[1] I. Bayati, S. Muggiasca, and A. Vandone. Experimental and numerical wind tunnel investigation of the aerodynamics of upwind soft sails. *Ocean Engineering*, 182(April):395–411, 2019. ISSN 00298018. doi: 10.1016/j.oceaneng.2019.04.037. URL https://doi.org/10.1016/j.oceaneng.2019.04.037.

[2] R. D. Blevins. Flow Induced Vibration by Robert D Blevins 2nd Ed, 1990.

[3] D. Boote and M. Caponetto. A Numerical Approach To The Design of Sailing Yacht Masts, 1991.

[4] S. C. Carlson. Catenary. URL https://www.britannica.com/science/catenary.

[5] E. Edge. Rectangular flat plate drag coefficient equation. URL https://www.engineersedge.com/fluid{_}flow/rectangular{_}flat{_}plate{_}drag{_}14036.htm.

[6] T. O. Foundation. The OpenFOAM 7 User Guide. (February):212, 2014.

[7] S. Ghelardi, A. Freda, C. M. Rizzo, and D. Villa. A Fluid-Structure Interaction case study on a square sail in a wind tunnel. *Ocean Engineering*, 163(May):136–147, 2018. ISSN 00298018. doi: 10.1016/j.oceaneng.2018.05.056. URL https://doi.org/10.1016/j.oceaneng.2018.05.056.

[8] S. GmbH. Post-processing strategies using paraview for formula student. URL https://youtu.be/ZxJnOB0Gwn8.

[9] P. Izaguirre-Alza. Numerical and Experimental Studies of Sail Aerodynamics. pages 1–282, 2012.

[10] K. Legursky. A modified model, simulation, and tests of a full-scale sailing yacht. *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*, pages 1–7, 2012. doi: 10.1109/OCEANS.2012.6404952.

[11] D. Li, Y. Zhang, P. Li, J. Dai, and G. Li. Aerodynamic performance of a new double-flap wing sail. *Polish Maritime Research*, 26(4):61–68, 2020. ISSN 20837429. doi: 10.2478/pomr-2019-0067.

[12] P. Łojek. Turbulence modeling in OpenFOAM. pages 1–5, 2019.

[13] R. Mohan, S. Sundararaj, and K. B. Thiagarajan. Numerical simulation of flow over buildings using OpenFOAM®. *AIP Conference Proceedings*, 2112(June), 2019. ISSN 15517616. doi: 10.1063/1.5112334.

[14] C. online. Turbulence free-stream boundary conditions, . URL https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions.

[15] C. online. Spalart-Allmaras turbulence model, . URL https://www.cfd-online.com/Wiki/Spalart-Allmaras_model.

[16] J. C. Puig. Chapter Appendix: Code for meshing an airfoil geometry. *OpenFOAM GUIDE*, pages 0–115, 2014. doi: 10.1176/appi.books.9780890423349.11451.

[17] D. Segersson. CFD with OpenSource software A tutorial to urban wind flow using. 2017.

[18] B. Sunden. Crossflow over tubes. URL http://thermopedia.com/content/1216/.

[19] A. Svirin. Equation of Catenary. URL https://www.math24.net/equation-catenary/.

[20] K. B. T. Critical reynold's number for turbulent external flow. URL https://www.quora.com/What-is-Reynoldss-number-for-turbulent-flow-in-external-flow.

[21] Wikipedia. Sail. URL https://en.wikipedia.org/wiki/Sail{#}History.